

CS 486/586 Intro Database Management Systems

Graduate Student Project - Final Deliverable

Team : Chethana Muppalam(cmuppa), Orobosa Ekhator (orobosa)

Subject: Tokyo 2021 Olympics Database

The modern Olympics have been held for several years as far back as the 1800s. The inspiration of this dates back to ancient greece. The Olympics is one of the leading international sports events that gathers thousands of athletes all around the world to participate in diverse sports. The Olympics are held every four years and alternated between Summer and Winter Olympics every two years within the four year period.

Having had a brief history of the Olympics, the idea behind creating a database for the Tokyo Olympics is to have an exhaustive compilation of all the activities that were held at the Olympics 2021 games. This includes the participating countries, names of athletes, coaches, medals earned, sports and other information. This database will be useful in educating the public/students on national games, improving sports research and also informing decision making. For example, the IOC(International Olympics committee) can use the above database in selecting host cities.

Potential use cases: The olympics database is a compilation that can be used to access information about different facets of the olympics games. It highlights the track records of games that were played and the results of the outcomes. This database can be used to acquire the knowledge of the last olympics events and view all the statistics of different games. This database can also be used as a source of data in performing predictive analytics in order to forecast the future of olympics games. This can be advantageous to sponsors or investors for the next coming year's olympics.

Data Sources and Discovery:

About sources of the Data: There are several sources from which the data was collected. Some of it include the data from olympics.com, olympics dataset on Kaggle and data world. The information sourced from these sources were systematically extracted, removed noisy/irrelevant data and positioned in .csv format. Subsequently, these CSV files serve as the conduit for populating the database.

Sources used in the project:

<https://www.kaggle.com/datasets/arjunprasadsarkhel/2021-olympics-in-tokyo>

<https://data.world/datasets/olympics>

<https://olympics.com/en/olympic-games/tokyo-2020>

Data Normalization and Validation:

- Organized the data to ensure consistency and conformity to predefined standards, reflecting the defined schema in the Tokyo Olympics database.
- Normalized the data to eliminate redundancy and improve data integrity.
- Validated the data to detect and rectify errors, inconsistencies, or inaccuracies in the collected information.
- Ensured that data types, constraints, and relationships align with the specifications of the olympics database schema.

Conversion to CSV and Cleanup:

- Converted the normalized and validated Tokyo Olympics data into CSV format, where each CSV file corresponds to a specific table in the database schema.
- Performed additional cleanup tasks during the conversion process, addressing issues like missing values, formatting discrepancies, or anomalies in the data.
- The resulting CSV files were well-structured and adhered to the schema of the Tokyo Olympics database.

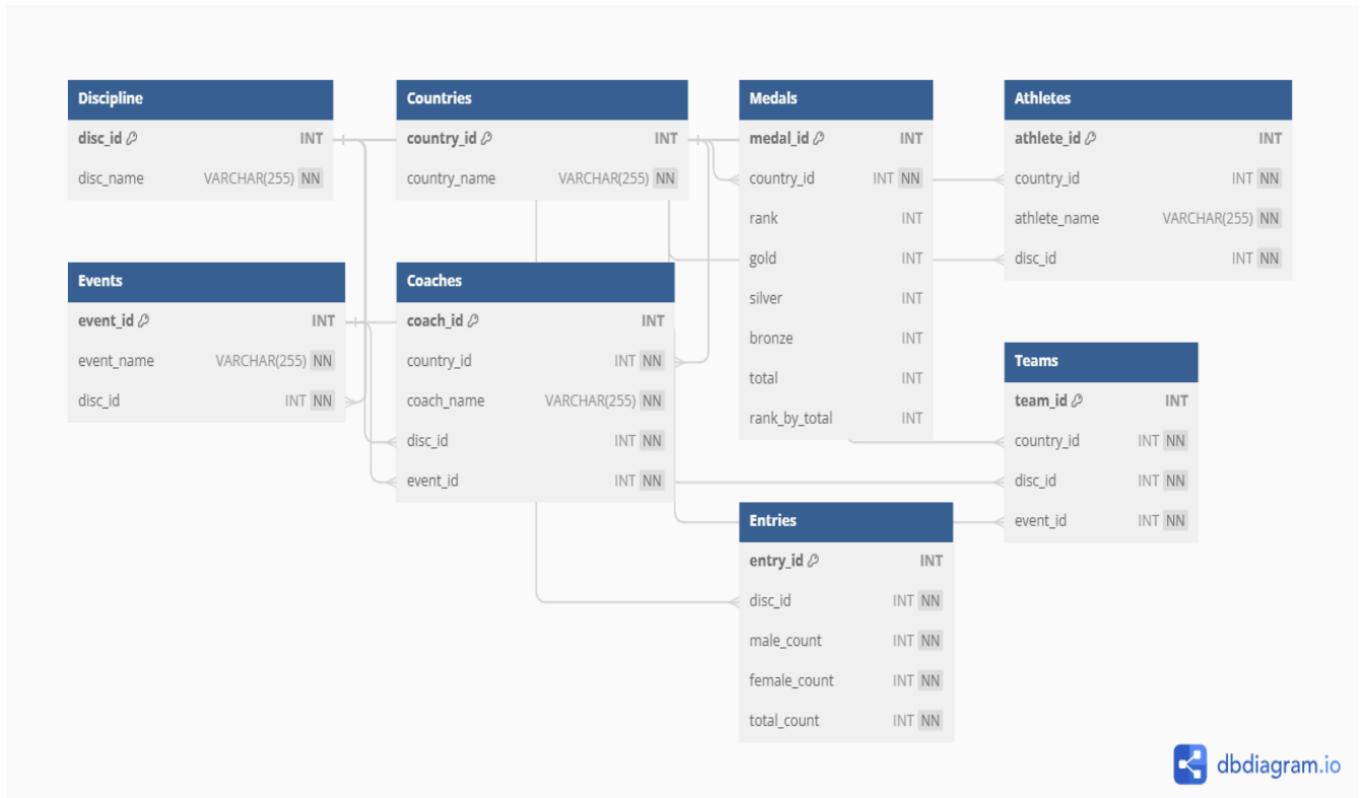
Loading CSV Files into the Database:

- Used SQL commands to create tables within the Tokyo Olympics database.
- Used Surrogate Auto Generated Keys to uniquely identify the record in the tables designed.
- Created temporary tables to load the data from csv to temporary tables.
- The formatted data from the CSV files were loaded into their respective temporary tables in the database.

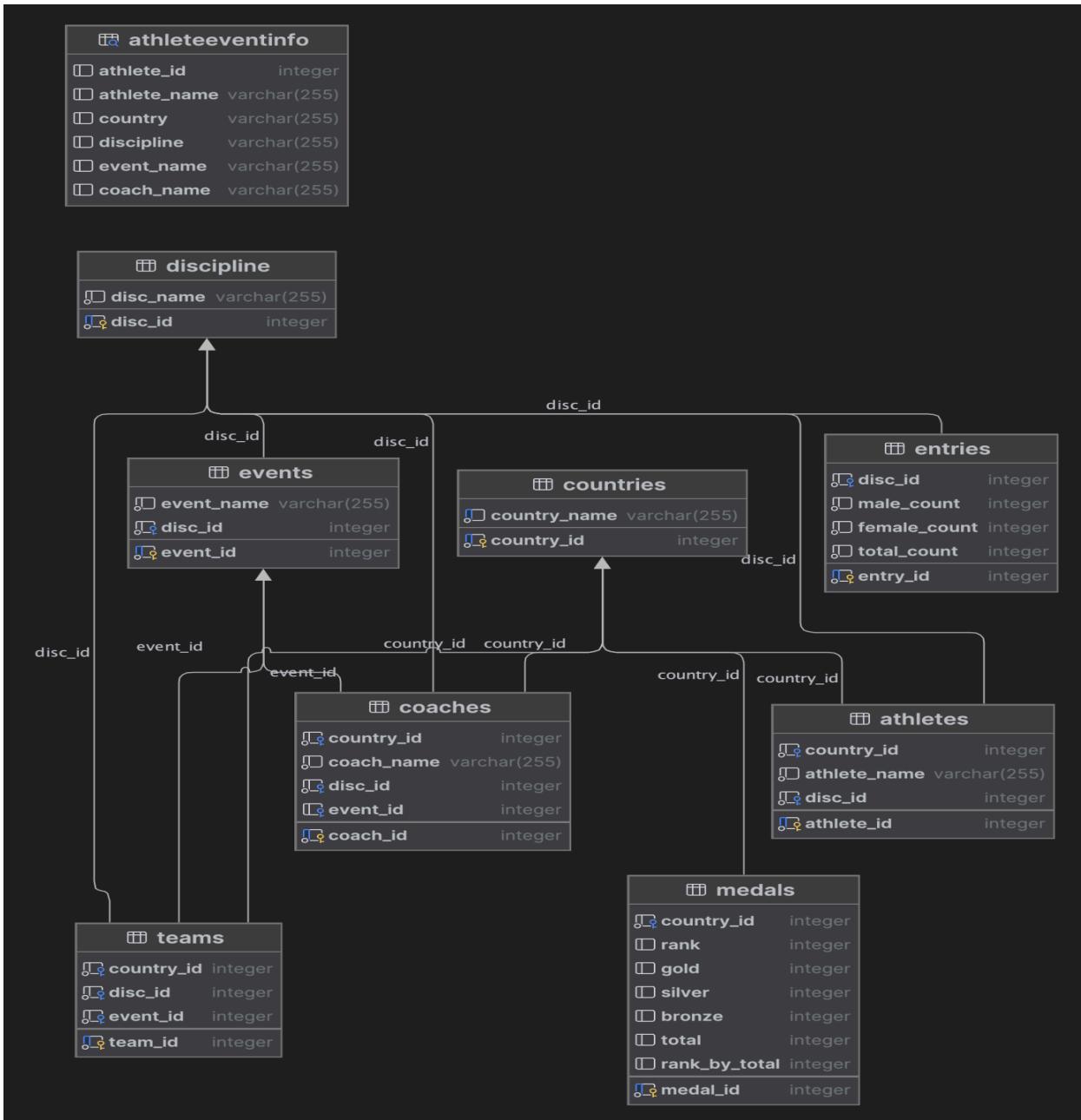
```
Command used: \copy tokyo_olympics.Table_Name(Column_name/s)
FROM '/u/chetha/dbms_olympics_project/CSV_FILE_NAME.csv' DELIMITER ','
CSV HEADER;
```

- Extracted the necessary records and columns from the temporary table and fed into the designed database table.
- Dropped the temporary table after data loading.
- Ensured that the relationships between tables are maintained, foreign keys are appropriately linked and created indexes.
- Conduct thorough checks to confirm the successful transfer of data, and verify that the database now reflects the organized and validated Tokyo Olympics information.

ER Diagram:



Relational Schema:



Database Design

Schema:

```
DROP SCHEMA IF EXISTS tokyo_olympics CASCADE;  
  
CREATE SCHEMA tokyo_olympics;  
  
SET search_path TO tokyo_olympics;
```

Tables:

Discipline

The table ‘Discipline’ in the ‘Tokyo Olympics’ database represents information about different disciplines or sports in the Tokyo Olympics. Here’s a description for each column:

- **disc_id(INT generated by default as identity primary key)** This column serves as the primary key for the table. It is of type INT (integer) and is automatically generated by the database system, ensuring that each entry in the table has a unique identifier.
- **disc_name (VARCHAR(255) NOT NULL)** This column stores the name of each discipline or sport in the Tokyo Olympics. It is of type VARCHAR(255), allowing for variable-length character strings of up to 255 characters. The NOT NULL constraint ensures that a value must be provided for this column, meaning each discipline entry must have a name.

```
CREATE TABLE Discipline (  
  
disc_id INT generated by default as identity primary key,  
  
disc_name VARCHAR(255) NOT NULL  
);
```

A screenshot of a database interface showing the result of a SQL query. The query is:

```
12 ✓ select count(*) from tokyo_olympics.Discipline;
```

The result is displayed in a table with one row:

	count
1	46

A screenshot of a database interface showing the result of a SQL query. The query is:

```
select * from tokyo_olympics.Discipline LIMIT 5;
```

The result is displayed in a table with 5 rows:

disc_id	disc_name
1	3x3 Basketball
2	Archery
3	Artistic Gymnastics
4	Artistic Swimming
5	Athletics

Countries

The table ‘Countries’ in the ‘Tokyo Olympics’ database represents information about different countries that participated in the Tokyo Olympics. Here’s a description for each column:

- **country_id(INT generated by default as identity primary key)** This column serves as the primary key for the table. It is of type INT (integer) and is automatically generated by the database system, ensuring that each entry in the table has a unique identifier.
- **country_name (VARCHAR(255) NOT NULL)** This column stores the name of each country that participated in the Tokyo Olympics. It is of type VARCHAR(255), allowing for variable-length character strings of up to 255 characters. The NOT NULL constraint ensures that a value must be provided for this column, meaning each country entry must have a name.
- **CONSTRAINT unique_country_name UNIQUE(country_name)** This constraint ensures that the values in the ‘country_name’ column must be unique across all rows in the table. It prevents duplicate country names from being inserted into the table.

```

CREATE TABLE Countries (
    country_id INT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
    country_name VARCHAR(255) NOT NULL,
    CONSTRAINT unique_country_name UNIQUE (country_name)
);

```

The screenshot shows a database interface with two panes. The top pane contains the SQL query: `select count(*) from tokyo_olympics.Countries;`. The bottom pane displays the results in a table format. The table has one row with the value 93 under the column labeled 'count'.

count
93

The screenshot shows a database interface with two panes. The top pane contains the SQL query: `select * from tokyo_olympics.Countries LIMIT 5;`. The bottom pane displays the results in a table format. The table has five rows, each with a 'country_id' and a 'country_name'. The data is as follows:

country_id	country_name
1	Argentina
2	Armenia
3	Australia
4	Austria
5	Azerbaijan

Medals

The table "Medals" stores information about the medal count of each country in the Olympics, including the counts of gold, silver, and bronze medals, as well as the total medal count and the ranking based on total medals won. The table is linked to the "Countries" table through a foreign key constraint. Here's a description for each column:

- **medal_id(INT generated by default as identity primary key)** : This column serves as the primary key for the table. It is of type INT (integer) and is automatically generated by the database system. Each entry in the "Medals" table is assigned a unique identifier to distinguish it from others.
- **country_id(INT NOT NULL)**: This column stores the identifier of the country associated with the medal count. It is of type INT and is marked as NOT NULL, meaning that a valid country identifier must be provided for each entry.

FOREIGN KEY (country_id) REFERENCES Countries(country_id): This line establishes a foreign key constraint on the country_id column, referencing the country_id column in the "Countries" table. It ensures that the country_id values stored in the "Medals" table correspond to valid country identifiers present in the "Countries" table.

- **rank INT** : This column stores the rank of the country based on its total medal count. It represents the position of the country relative to others in terms of the number of medals won.
- **gold(INT, silver INT, bronze INT)**: These columns store the counts of gold, silver, and bronze medals won by the country, respectively. Each column is of type INT and represents the number of medals of each type won by the country.
- **total INT**: This column stores the total number of medals won by the country, summing up the counts of gold, silver, and bronze medals.
- **rank_by_total INT**: This column stores the rank of the country based on its total medal count. Similar to the rank column, it represents the position of the country relative to others based on the total number of medals won.

```
CREATE TABLE Medals (
    medal_id INT generated by default as identity primary key,
    country_id INT NOT NULL,
    rank INT,
    gold INT,
    silver INT,
    bronze INT,
    total INT,
    rank_by_total INT,
```

```
FOREIGN KEY (country_id) REFERENCES Countries(country_id)
);
```

A screenshot of a database interface showing a single row of results for a COUNT(*) query. The query is:

```
12 ✓ | select count(*) from tokyo_olympics.Medals;|  
13
```

The result table has one row with the following data:

	count
1	93

A screenshot of a database interface showing the top 5 rows of the Medals table. The query is:

```
select * from tokyo_olympics.Medals LIMIT 5;
```

The result table has 5 rows of data:

medal_id	country_id	rank	gold	silver	bronze	total	rank_by_total
1	1	91	1	39	41	33	113
2	2	65	2	38	32	18	88
3	3	46	3	27	14	17	58
4	4	34	4	22	21	22	65

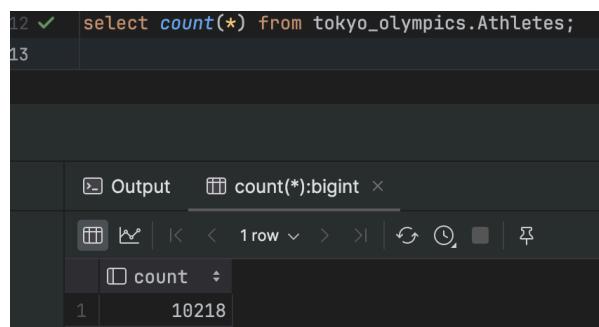
Athletes

The table ‘Athletes’ in the ‘Tokyo Olympics’ database represents information about different athletes that participated in the Tokyo Olympics. Here’s a description for each column:

- **athlete_id(INT generated by default as identity primary key)** This column serves as the primary key for the table. It is of type INT (integer) and is automatically generated by the database system, ensuring that each entry in the table has a unique identifier.
- **country_id (INT NOT NULL):** This column stores the identifier of the country associated with the athlete. It is of type INT and is marked as NOT NULL, meaning that a valid country identifier must be provided for each entry.
- **athlete_name (VARCHAR(255) NOT NULL)** This column stores the name of each athlete that participated in the Tokyo Olympics. It is of type VARCHAR(255), allowing for variable-length character strings of up to 255 characters. The NOT NULL constraint ensures that a value must be provided for this column, meaning each athlete entry must have a name.

- **disc_id (INT NOT NULL):** This column stores the identifier of the discipline associated with the athlete. It is of type INT and is marked as NOT NULL, meaning that a valid discipline identifier must be provided for each entry.
- **FOREIGN KEY (country_id) REFERENCES Countries(country_id):** This line establishes a foreign key constraint on the country_id column, referencing the country_id column in the "Countries" table. It ensures that the country_id values stored in the "Athletes" table correspond to valid country identifiers present in the "Countries" table.
- **FOREIGN KEY (disc_id) REFERENCES Discipline(disc_id):** This line establishes a foreign key constraint on the disc_id column, referencing the disc_id column in the "Discipline" table. It ensures that the disc_id values stored in the "Athletes" table correspond to valid discipline identifiers present in the "Discipline" table.

```
CREATE TABLE Athletes (
    athlete_id INT generated by default as identity primary key,
    country_id INT NOT NULL,
    athlete_name VARCHAR(255) NOT NULL,
    disc_id INT NOT NULL,
    FOREIGN KEY (country_id) REFERENCES Countries(country_id),
    FOREIGN KEY (disc_id) REFERENCES Discipline(disc_id)
);
```



The screenshot shows a database query results window. The query is:

```
12 ✓ | select count(*) from tokyo_olympics.Athletes;
13
```

The results table has one row:

count	10218
1	10218

The screenshot shows a MySQL Workbench interface with the following details:

- Query Editor:** The query is `select * from tokyo_olympics.Athletes LIMIT 5;`.
- Output Window:** The results are displayed in a table titled "win2024adb0046.tokyo_olympics.athletes".
- Table Structure:**

	athlete_id	country_id	athlete_name	disc_id
1	1	64	AALERUD Katrine	16
2	2	81	ABAD Nestor	3
3	3	44	ABAGNALE Giovanni	30
4	4	81	ABALDE Alberto	8
5	5	81	ABALDE Tamara	8

Events

The table 'Events' in the 'Tokyo Olympics' database represents information about different events in the Tokyo Olympics. Here's a description for each column:

- **event_id(INT generated by default as identity primary key)** This column serves as the primary key for the table. It is of type INT (integer) and is automatically generated by the database system, ensuring that each entry in the table has a unique identifier.
- **event_name (VARCHAR(255) NOT NULL)** This column stores the name of each event in the Tokyo Olympics. It is of type VARCHAR(255), allowing for variable-length character strings of up to 255 characters. The NOT NULL constraint ensures that a value must be provided for this column, meaning each event entry must have a name.
- **disc_id (INT NOT NULL):** This column stores the identifier of the discipline associated with the event. It is of type INT and is marked as NOT NULL, meaning that a valid discipline identifier must be provided for each entry.
- **FOREIGN KEY (disc_id) REFERENCES Discipline(disc_id):** This line establishes a foreign key constraint on the disc_id column, referencing the disc_id column in the "Discipline" table. It ensures that the disc_id values stored in the "Events" table correspond to valid discipline identifiers present in the "Discipline" table.

```
CREATE TABLE Events (
    event_id INT generated by default as identity primary key,
    event_name VARCHAR(255) NOT NULL,
    disc_id INT NOT NULL,
    FOREIGN KEY (disc_id) REFERENCES Discipline(disc_id)
);
```

```

12 ✓ select count(*) from tokyo_olympics.Events;
13

Output count(*):bigint ×
count ↴
1 54

```

Output win2024adb0046.tokyo_olympics.events		
event_id	event_name	disc_id
1	Women	1
2	Men	1
3	Women's Team	2
4	Mixed Team	2
5	Men's Team	2

Coaches

The table ‘Coaches’ in the ‘Tokyo Olympics’ database represents information about different coaches that participated in the Tokyo Olympics. Here’s a description for each column:

- **coach_id(INT generated by default as identity primary key)** This column serves as the primary key for the table. It is of type INT (integer) and is automatically generated by the database system, ensuring that each entry in the table has a unique identifier.
- **country_id (INT NOT NULL):** This column stores the identifier of the country associated with the coaches. It is of type INT and is marked as NOT NULL, meaning that a valid country identifier must be provided for each entry.
- **coach_name (VARCHAR(255) NOT NULL)** This column stores the name of each coach that participated in the Tokyo Olympics. It is of type VARCHAR(255), allowing for variable-length character strings of up to 255 characters. The NOT NULL constraint ensures that a value must be provided for this column, meaning each coach entry must have a name.

- **disc_id (INT NOT NULL):** This column stores the identifier of the discipline associated with the coaches. It is of type INT and is marked as NOT NULL, meaning that a valid discipline identifier must be provided for each entry.
- **event_id (INT NOT NULL):** This column stores the identifier of the events associated with the coaches. It is of type INT and is marked as NOT NULL, meaning that a valid discipline identifier must be provided for each entry.
- **FOREIGN KEY (country_id) REFERENCES Countries(country_id):** This line establishes a foreign key constraint on the country_id column, referencing the country_id column in the "Countries" table. It ensures that the country_id values stored in the "Coaches" table correspond to valid country identifiers present in the "Countries" table.
- **FOREIGN KEY (disc_id) REFERENCES Discipline(disc_id):** This line establishes a foreign key constraint on the disc_id column, referencing the disc_id column in the "Discipline" table. It ensures that the disc_id values stored in the "Coaches" table correspond to valid discipline identifiers present in the "Discipline" table.
- **FOREIGN KEY (event_id) REFERENCES Events(event_id):** This line establishes a foreign key constraint on the event_id column, referencing the event_id column in the "Events" table. It ensures that the event_id values stored in the "Coaches" table correspond to valid event identifiers present in the "Events" table.

```
CREATE TABLE Coaches (
    coach_id INT generated by default as identity primary key,
    country_id INT NOT NULL,
    coach_name VARCHAR(255) NOT NULL,
    disc_id INT NOT NULL,
    event_id INT,
    FOREIGN KEY (country_id) REFERENCES Countries(country_id),
    FOREIGN KEY (disc_id) REFERENCES Discipline(disc_id),
    FOREIGN KEY (event_id) REFERENCES Events(event_id)
);
```

```

12 ✓ | select count(*) from tokyo_olympics.Coaches;
13 |

```

Output count(*):bigint

	count
1	1704

```

select * from tokyo_olympics.Coaches LIMIT 5;

```

cipline > disc_id

Output win2024adb0046.tokyo_olympics.coaches

coach_id	country_id	coach_name	disc_id	event_id
1	1	MIHAILOVIC Predrag	44	1
2	2	PARADELO David	44	1
3	3	BIRO Attila	44	1
4	4	MOTOMIYA Makihiro	44	1
5	5	HAVENGA Arno	44	1

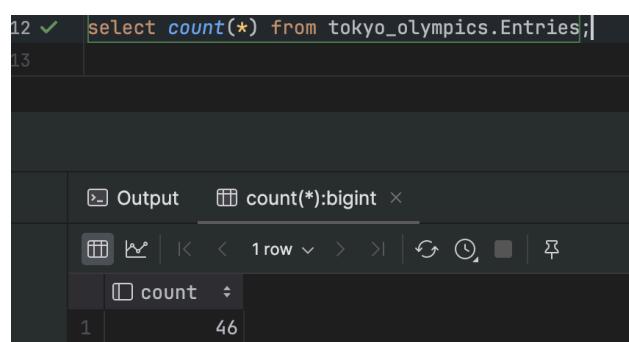
Entries

The table ‘Entries’ in the ‘Tokyo Olympics’ database represents information about different entries in the Tokyo Olympics. Here’s a description for each column:

- **entry_id(INT generated by default as identity primary key)** This column serves as the primary key for the table. It is of type INT (integer) and is automatically generated by the database system, ensuring that each entry in the table has a unique identifier.
- **disc_id (INT NOT NULL):** This column stores the identifier of the discipline associated with the entries. It is of type INT and is marked as NOT NULL, meaning that a valid discipline identifier must be provided for each entry.
- **male_count (INT NOT NULL):** This column stores the number of male athletes that participated in the Tokyo Olympics . It is of type INT and is marked as NOT NULL, meaning that a valid number of male athletes must be provided for each entry.
- **female_count (INT NOT NULL):** This column stores the number of female athletes that participated in the Tokyo Olympics.. It is of type INT and is marked as NOT NULL, meaning that a valid number of female athletes must be provided for each entry.

- **total_count (INT NOT NULL):** This column stores the total number of males and female athletes that participated in the Tokyo Olympics. It is of type INT and is marked as NOT NULL, meaning that a valid number of male and female athletes must be provided for each entry.
- **FOREIGN KEY (disc_id) REFERENCES Discipline(disc_id):** This line establishes a foreign key constraint on the disc_id column, referencing the disc_id column in the "Discipline" table. It ensures that the disc_id values stored in the "Entries" table correspond to valid discipline identifiers present in the "Discipline" table.

```
CREATE TABLE Entries (
    entry_id INT generated by default as identity primary key,
    disc_id INT NOT NULL,
    male_count INT NOT NULL,
    female_count INT NOT NULL,
    total_count INT NOT NULL,
    FOREIGN KEY (disc_id) REFERENCES Discipline(disc_id)
);
```



The screenshot shows a database interface with a query editor and a results table. The query in the editor is:

```
12 ✓ | select count(*) from tokyo_olympics.Entries;|
```

The results table has one row with the following data:

	count(*)
1	46

```
select * from tokyo_olympics.Entries LIMIT 5;
```

entry_id	disc_id	male_count	female_count	total_count
1	1	32	32	64
2	2	64	64	128
3	3	98	98	196
4	4	0	105	105
5	5	1072	969	2041

Teams

The table 'Teams' in the 'Tokyo Olympics' database represents information about different teams that participated in the Tokyo Olympics. Here's a description for each column:

- **team_id(INT generated by default as identity primary key)**: This column serves as the primary key for the table. It is of type INT (integer) and is automatically generated by the database system, ensuring that each entry in the table has a unique identifier.
- **country_id (INT NOT NULL)**: This column stores the identifier of the country associated with the teams. It is of type INT and is marked as NOT NULL, meaning that a valid country identifier must be provided for each entry.
- **disc_id (INT NOT NULL)**: This column stores the identifier of the discipline associated with the teams. It is of type INT and is marked as NOT NULL, meaning that a valid discipline identifier must be provided for each entry.
- **event_id (INT NOT NULL)**: This column stores the identifier of the events associated with the teams. It is of type INT and is marked as NOT NULL, meaning that a valid discipline identifier must be provided for each entry.
- **FOREIGN KEY (country_id) REFERENCES Countries(country_id)**: This line establishes a foreign key constraint on the country_id column, referencing the country_id column in the "Countries" table. It ensures that the country_id values stored in the "Teams" table correspond to valid country identifiers present in the "Countries" table.
- **FOREIGN KEY (disc_id) REFERENCES Discipline(disc_id)**: This line establishes a foreign key constraint on the disc_id column, referencing the disc_id column in the "Discipline" table. It ensures that the disc_id values stored in the "Teams" table correspond to valid discipline identifiers present in the "Discipline" table.

- **FOREIGN KEY (event_id) REFERENCES Events(event_id):** This line establishes a foreign key constraint on the event_id column, referencing the event_id column in the "Events" table. It ensures that the event_id values stored in the "Teams" table correspond to valid event identifiers present in the "Events" table.

```
CREATE TABLE Teams (
    team_id INT generated by default as identity primary key,
    country_id INT NOT NULL,
    disc_id INT NOT NULL,
    event_id INT NOT NULL,
    FOREIGN KEY (country_id) REFERENCES Countries(country_id),
    FOREIGN KEY (disc_id) REFERENCES Discipline(disc_id),
    FOREIGN KEY (event_id) REFERENCES Events(event_id)
);
```

```
12 ✓ select count(*) from tokyo_olympics.Teams;
13
```

Output count(*):bigint ×

1 row

	count
1	583

```
select * from tokyo_olympics.Teams LIMIT 5;
```

	team_id	country_id	disc_id	event_id
1	1	30	1	1
2	2	44	1	1
3	3	46	1	1
4	4	57	1	1
5	5	73	1	1

Index declarations:

```
CREATE INDEX tab_ix ON Discipline(disc_id);
```

```
CREATE INDEX count_ix ON Countries(country_id)
```

```
CREATE INDEX med_ix ON Medals(medal_id);
```

```
CREATE INDEX ath_ix ON Athletes(athlete_id);
```

```
CREATE INDEX evt_ix ON Events(event_id);
```

```
CREATE INDEX coach_ix ON Coaches(coach_id);
```

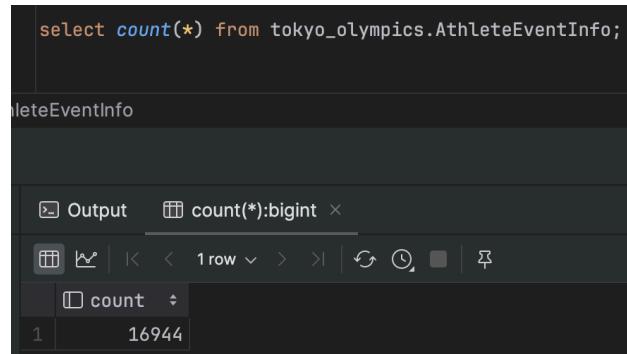
```
CREATE INDEX ent_ix ON Entries(entry_id);
```

```
CREATE INDEX team_ix ON Teams(team_id);
```

View:

The table below is a view called ‘AthleteEventInfo’ in the ‘Tokyo Olympics’ database that combines 6 tables together in order to access information. This is a virtual table.

```
CREATE OR REPLACE VIEW tokyo_olympics.AthleteEventInfo AS
SELECT a.athlete_id, a.athlete_name, c.country_name AS country, d.disc_name AS
discipline, e.event_name,
co.coach_name
FROM tokyo_olympics.Athletes a
INNER JOIN tokyo_olympics.Countries c ON a.country_id = c.country_id
INNER JOIN tokyo_olympics.Discipline d ON a.disc_id = d.disc_id
LEFT JOIN tokyo_olympics.Teams t ON a.country_id = t.country_id AND a.disc_id
= t.disc_id
LEFT JOIN tokyo_olympics.Events e ON t.event_id = e.event_id
LEFT JOIN tokyo_olympics.Coaches co ON t.country_id = co.country_id AND
t.disc_id = co.disc_id AND t.event_id = co.event_id
ORDER BY a.athlete_id;
```

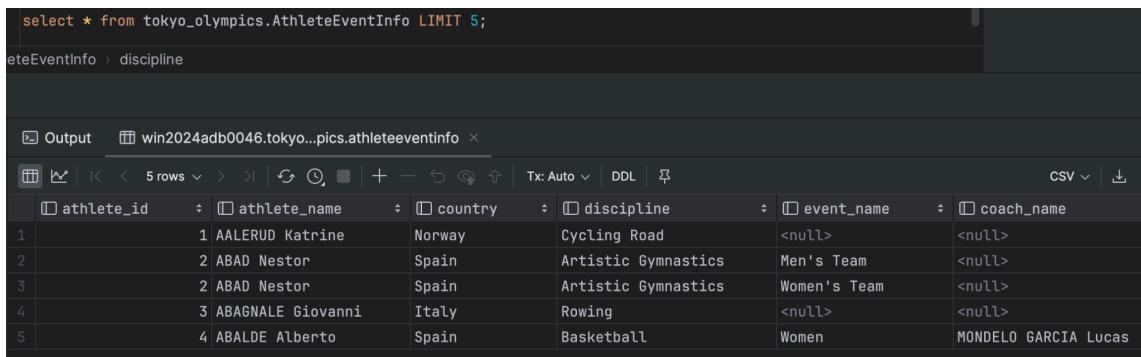


A screenshot of a database interface showing the results of a query. The query is:

```
select count(*) from tokyo_olympics.AthleteEventInfo;
```

The results are displayed in a table with one row:

count	16944
1	16944



A screenshot of a database interface showing the results of a query. The query is:

```
select * from tokyo_olympics.AthleteEventInfo LIMIT 5;
```

The results are displayed in a table with 5 rows:

athlete_id	athlete_name	country	discipline	event_name	coach_name
1	AALERUD Katrine	Norway	Cycling Road	<null>	<null>
2	ABAD Nestor	Spain	Artistic Gymnastics	Men's Team	<null>
2	ABAD Nestor	Spain	Artistic Gymnastics	Women's Team	<null>
3	ABAGNALE Giovanni	Italy	Rowing	<null>	<null>
4	ABALDE Alberto	Spain	Basketball	Women	MONDELO GARCIA Lucas

Querying from the Database:

- How many athletes from each country whose names begin with 'A' participated, and how many medals did they win for their country?

```
SELECT c.country_name, COUNT(DISTINCT a.athlete_id) AS num_athletes,
SUM(m.total) AS total_medals
FROM tokyo_olympics.Athletes a
JOIN tokyo_olympics.Countries c ON a.country_id = c.country_id
LEFT JOIN tokyo_olympics.Medals m ON a.country_id = m.country_id
WHERE a.athlete_name LIKE 'A%'
GROUP BY c.country_name;
```

The screenshot shows a database interface with a query editor containing the SQL code and a results table below it.

country_name	num_athletes	total_medals
Argentina	9	27
Armenia	3	12
Australia	19	874
Austria	5	35
Azerbaijan	8	56
Bahamas	1	2
Bahrain	9	9
Belarus	2	14
Belgium	3	21
Bermuda	1	1
Botswana	1	1
Brazil	22	462
Bulgaria	3	18
Canada	12	288

- Which discipline has the event with the most participating countries and the highest average medal count (excluding gold) per athlete?

```
SELECT d.disc_name AS Discipline_Name, e.event_name AS Event_Name,
COUNT(DISTINCT t.country_id) AS Participating_Countries,
AVG(CASE WHEN m.gold = 0 THEN m.silver + m.bronze ELSE NULL END) AS Avg_Medal_Count_Per_Athlete
FROM tokyo_olympics.Events e
JOIN tokyo_olympics.Teams t ON e.event_id = t.event_id
JOIN tokyo_olympics.Discipline d ON e.disc_id = d.disc_id
LEFT JOIN tokyo_olympics.Medals m ON t.country_id = m.country_id
GROUP BY d.disc_name, e.event_name
ORDER BY COUNT(DISTINCT t.country_id) DESC,
AVG(CASE WHEN m.gold = 0 THEN m.silver + m.bronze ELSE NULL END) DESC LIMIT 1;
```

```

SELECT d.disc_name AS Discipline_Name, e.event_name AS Event_Name, COUNT(DISTINCT t.country_id) AS Participating_Countries,
AVG(CASE WHEN m.gold = 0 THEN m.silver + m.bronze ELSE NULL END) AS Avg_Medal_Count_Per_Athlete
FROM tokyo_olympics.Events e
JOIN tokyo_olympics.Teams t ON e.event_id = t.event_id
JOIN tokyo_olympics.Discipline d ON e.disc_id = d.disc_id
LEFT JOIN tokyo_olympics.Medals m ON t.country_id = m.country_id
GROUP BY d.disc_name, e.event_name
ORDER BY COUNT(DISTINCT t.country_id) DESC,
AVG(CASE WHEN m.gold = 0 THEN m.silver + m.bronze ELSE NULL END) DESC LIMIT 1;

```

Output Result 2

discipline_name	event_name	participating_countries	avg_medal_count_per_athlete
Archery	Mixed Team	24	3.75

3. List all athletes and their respective countries participating in all kinds of Basketball discipline and event type of the "Men".

```

SELECT Distinct a.athlete_name, c.country_name, d.disc_name
FROM tokyo_olympics.Athletes a
JOIN tokyo_olympics.Countries c ON a.country_id = c.country_id
JOIN tokyo_olympics.Teams t ON a.country_id = t.country_id
JOIN tokyo_olympics.Events e ON t.event_id = e.event_id
JOIN tokyo_olympics.Discipline d ON t.disc_id = d.disc_id
WHERE e.event_name = 'Men' AND d.disc_name like '%Basketball%';

```

```

SELECT Distinct a.athlete_name, c.country_name, d.disc_name
FROM tokyo_olympics.Athletes a
JOIN tokyo_olympics.Countries c ON a.country_id = c.country_id
JOIN tokyo_olympics.Teams t ON a.country_id = t.country_id
JOIN tokyo_olympics.Events e ON t.event_id = e.event_id
JOIN tokyo_olympics.Discipline d ON t.disc_id = d.disc_id
WHERE e.event_name = 'Men' AND d.disc_name like '%Basketball%';

```

Output Result 3

athlete_name	country_name	disc_name
1 ABAD Nestor	Spain	Basketball
2 ABAGNALE Giovanni	Italy	Basketball
3 ABALDE Alberto	Spain	Basketball
4 ABALDE Tamara	Spain	Basketball
5 ABALO Luc	France	Basketball
6 ABBINGH Lois	Netherlands	3x3 Basketball
7 ABBOT Emily	Australia	Basketball
8 ABDELWAHED Ahmed	Italy	Basketball
9 ABDI Bashir	Belgium	3x3 Basketball
10 ABDULJABBAR Ammar Riad	Germany	Basketball
11 ABE Hifumi	Japan	3x3 Basketball
12 ABE Hifumi	Japan	Basketball
13 ABE Takatoshi	Japan	3x3 Basketball
14 ABE Takatoshi	Japan	Basketball

4. List countries, along with their total gold medals and corresponding disciplines, sorted by gold medals from highest to lowest.

```

SELECT      c.country_name,      SUM(m.gold)      AS      total_gold_medals,
STRING_AGG(d.disc_name, ', ') AS disciplines_with_gold
FROM tokyo_olympics.Medals m
JOIN tokyo_olympics.Countries c ON m.country_id = c.country_id
JOIN tokyo_olympics.Teams t ON m.country_id = t.country_id
JOIN tokyo_olympics.Events e ON t.event_id = e.event_id
JOIN tokyo_olympics.Discipline d ON e.disc_id = d.disc_id WHERE m.gold > 0
GROUP BY c.country_name ORDER BY total_gold_medals DESC;
    
```

	country_name	total_gold_medals	disciplines_with_gold
1	Japan	1188	3x3 Basketball, 3x3 Basketball, Archery, Archery, ...
2	ROC	640	3x3 Basketball, 3x3 Basketball, Archery, Archery, ...
3	Great Britain	616	Archery, Archery, Archery, Artistic Gymnastics, Ar...
4	Australia	527	Archery, Archery, Artistic Swimming, Artistic Swi...
5	Italy	340	3x3 Basketball, Archery, Archery, Artistic Gymnast...
6	France	330	3x3 Basketball, Archery, Archery, Artistic Gymnast...
7	Germany	330	Archery, Archery, Artistic Gymnastics, Artistic Gy...
8	Netherlands	240	3x3 Basketball, Archery, Archery, Artistic Gymnast...
9	Canada	189	Archery, Artistic Gymnastics, Artistic Swimming, A...
10	Brazil	147	Archery, Artistic Gymnastics, Athletics, Athletics...
11	Republic of Korea	96	Archery, Archery, Archery, Basketball, Fencing, Fe...
12	New Zealand	91	Cycling Track, Cycling Track, Cycling Track, Cycli...
13	Hungary	78	Fencing, Fencing, Fencing, Handball, Swimming, Swi...
14	Poland	76	3x3 Basketball, Archery, Athletics, Athletics, Ath...

5. List the athletes who participated without assigned coaches, along with their respective disciplines and countries.

```

SELECT      a.athlete_name,      d.disc_name,      c.country_name      FROM
tokyo_olympics.Athletes a
JOIN tokyo_olympics.Discipline d ON a.disc_id = d.disc_id
JOIN tokyo_olympics.Countries c ON a.country_id = c.country_id
WHERE NOT EXISTS (
SELECT 1 FROM tokyo_olympics.Coaches WHERE
Coaches.country_id = a.country_id AND Coaches.disc_id = a.disc_id
);
    
```

```

SELECT a.athlete_name, d.disc_name, c.country_name FROM tokyo_olympics.Athletes a
JOIN tokyo_olympics.Discipline d ON a.disc_id = d.disc_id
JOIN tokyo_olympics.Countries c ON a.country_id = c.country_id
WHERE NOT EXISTS (
    SELECT 1 FROM tokyo_olympics.Coaches WHERE
    Coaches.country_id = a.country_id AND Coaches.disc_id = a.disc_id
);

```

Output Result 6 ×

The screenshot shows a table with three columns: athlete_name, disc_name, and country_name. The data is as follows:

	athlete_name	disc_name	country_name
1	AALERUD Katrine	Cycling Road	Norway
2	ABAD Nestor	Artistic Gymnastics	Spain
3	ABAGNALE Giovanni	Rowing	Italy
4	ABBASALI Hamideh	Karate	Islamic Republic of Iran
5	ABBASOV Islam	Wrestling	Azerbaijan
6	ABBOT Emily	Rhythmic Gymnastics	Australia
7	ABDALLA Abubaker Haydar	Athletics	Qatar
8	ABDEL LATIF Radwa	Shooting	Egypt
9	ABDEL RAZEK Samy	Shooting	Egypt
10	ABDELAZIZ Abdalla	Karate	Egypt
11	ABDELAZIZ Farah	Table Tennis	Egypt
12	ABDELAZIZ Feryal	Karate	Egypt
13	ABDELMAWGoud Mohamed	Judo	Egypt
14	ABDELMOTTALEB Diaeldin Kamal Gouda	Wrestling	Egypt

6. List Top 10 disciplines with the most participating countries.

```

SELECT d.disc_name, COUNT(DISTINCT c.country_id) AS participating_countries
FROM tokyo_olympics.Entries en
JOIN tokyo_olympics.Discipline d ON en.disc_id = d.disc_id
JOIN tokyo_olympics.Athletes a ON en.disc_id = a.disc_id
JOIN tokyo_olympics.Countries c ON a.country_id = c.country_id
GROUP BY d.disc_name
ORDER BY participating_countries DESC
LIMIT 10;

```

```

SELECT d.disc_name, COUNT(DISTINCT c.country_id) AS participating_countries
FROM tokyo_olympics.Entries en
JOIN tokyo_olympics.Discipline d ON en.disc_id = d.disc_id
JOIN tokyo_olympics.Athletes a ON en.disc_id = a.disc_id
JOIN tokyo_olympics.Countries c ON a.country_id = c.country_id
GROUP BY d.disc_name
ORDER BY participating_countries DESC
LIMIT 10;

```

Output Result 7 ×

	disc_name	participating_countries
1	Swimming	91
2	Athletics	91
3	Shooting	72
4	Judo	70
5	Rowing	61
6	Boxing	57
7	Cycling Road	54
8	Weightlifting	53
9	Wrestling	52
10	Artistic Gymnastics	50

7. What are the names of athletes from Spain who won the gold medal in the Artistic Gymnastics display along with the event's name in the Tokyo 2021 Olympics?

```

SELECT a.athlete_name, e.event_name FROM tokyo_olympics.Athletes a
JOIN tokyo_olympics.Medals m ON a.country_id = m.country_id
JOIN tokyo_olympics.Discipline d ON a.disc_id = d.disc_id
join tokyo_olympics.events e on d.disc_id = e.disc_id
WHERE
    m.gold > 0
    AND d.disc_name = 'Artistic Gymnastics'
    AND a.country_id = (SELECT country_id FROM tokyo_olympics.Countries
    WHERE country_name = 'Spain');

```

```

SELECT a.athlete_name, e.event_name FROM tokyo_olympics.Athletes a
JOIN tokyo_olympics.Medals m ON a.country_id = m.country_id
JOIN tokyo_olympics.Discipline d ON a.disc_id = d.disc_id
join tokyo_olympics.events e on d.disc_id = e.disc_id
WHERE
    m.gold > 0
    AND d.disc_name = 'Artistic Gymnastics'
    AND a.country_id = (SELECT country_id FROM tokyo_olympics.Countries
    WHERE country_name = 'Spain');

```

Output Result 8 ×

The screenshot shows a database query results window with the title "Output Result 8 ×". Below the title are standard navigation buttons for a table: back, forward, search, and refresh. The table has two columns: "athlete_name" and "event_name". The data consists of 18 rows, numbered 1 to 14. Rows 1 through 6 show pairs of athletes and their respective teams. Rows 7 through 12 show pairs of women's team members. Rows 13 and 14 show pairs of men's team members.

	athlete_name	event_name
1	ABAD Nestor	Men's Team
2	ABAD Nestor	Women's Team
3	BECHDEJU Laura	Men's Team
4	BECHDEJU Laura	Women's Team
5	DIALLO Thierno	Men's Team
6	DIALLO Thierno	Women's Team
7	GONZALEZ Marina	Men's Team
8	GONZALEZ Marina	Women's Team
9	MIR Nicolau	Men's Team
10	MIR Nicolau	Women's Team
11	PETISCO Alba	Men's Team
12	PETISCO Alba	Women's Team
13	PLATA Joel	Men's Team
14	PLATA Joel	Women's Team

8. How many disciplines did Belgium compete in during the Tokyo 2021 Olympics?

```

select count(distinct d.disc_id) as num_disciplines
from tokyo_olympics.teams
join tokyo_olympics.countries on teams.country_id = countries.country_id
join tokyo_olympics.discipline d on d.disc_id = teams.disc_id
join tokyo_olympics.events on teams.event_id = events.event_id
where country_name = 'Belgium';

```

```
✓ select count(distinct d.disc_id) as num_disciplines
  from tokyo_olympics.teams
  join tokyo_olympics.countries on teams.country_id = countries.country_id
  join tokyo_olympics.discipline d on d.disc_id = teams.disc_id
  join tokyo_olympics.events on teams.event_id = events.event_id
  where country_name = 'Belgium';

Output num_disciplines:bigint ×
num_disciplines
1 | 7
```

9. What are the disciplines in Tokyo Olympics with a minimum number of 3 participating countries

```
select d.disc_name
from tokyo_olympics.discipline d
join tokyo_olympics.events e on d.disc_id = e.disc_id
join tokyo_olympics.teams t on e.event_id = t.event_id
join tokyo_olympics.athletes a on t.country_id = a.country_id and t.disc_id = a.disc_id
group by d.disc_name
having count(distinct t.country_id) >=3;
```

```

✓ select d.disc_name
  from tokyo_olympics.discipline d
  join tokyo_olympics.events e on d.disc_id = e.disc_id
  join tokyo_olympics.teams t on e.event_id = t.event_id
  join tokyo_olympics.athletes a on t.country_id = a.country_id and t.disc_id = a.disc_id
  group by d.disc_name
  having count(distinct t.country_id) >=3;

```

Output

disc_name
3x3 Basketball
Archery
Artistic Gymnastics
Artistic Swimming
Athletics
Basketball
Cycling Track
Fencing
Football
Handball
Hockey
Rhythmic Gymnastics

10. What is the rank achieved by Belgium in the 3x3 Basketball discipline?

```

SELECT m.rank, cn.country_name, d.disc_name FROM tokyo_olympics.Medals m
JOIN tokyo_olympics.Teams t ON m.country_id = t.country_id
JOIN tokyo_olympics.Countries cn ON m.country_id = cn.country_id
JOIN tokyo_olympics.Events e ON t.event_id = e.event_id
JOIN tokyo_olympics.Discipline d ON e.disc_id = d.disc_id
WHERE d.disc_name = '3x3 Basketball' AND t.country_id =
(SELECT country_id FROM tokyo_olympics.Countries WHERE country_name = 'Belgium');

```

```

✓ SELECT m.rank, cn.country_name, d.disc_name FROM tokyo_olympics.Medals m
  JOIN tokyo_olympics.Teams t ON m.country_id = t.country_id
  JOIN tokyo_olympics.Countries cn ON m.country_id = cn.country_id
  JOIN tokyo_olympics.Events e ON t.event_id = e.event_id
  JOIN tokyo_olympics.Discipline d ON e.disc_id = d.disc_id
  WHERE d.disc_name = '3x3 Basketball' AND t.country_id =
  (SELECT country_id FROM tokyo_olympics.Countries WHERE country_name = 'Belgium');


```

Output

rank	country_name	disc_name
1	29	Belgium
		3x3 Basketball

11. What are the unique names of coaches and discipline names for teams where total athlete count is greater than 10.

```
select distinct c.coach_name, d.disc_name from tokyo_olympics.coaches c
join tokyo_olympics.countries cn on c.country_id = cn.country_id
join tokyo_olympics.teams t on c.country_id = t.country_id and c.disc_id =
t.disc_id
join tokyo_olympics.entries e on t.disc_id = e.disc_id
join tokyo_olympics.discipline d on d.disc_id = t.disc_id
where e.total_count > 10;
```

✓ select distinct c.coach_name, d.disc_name from tokyo_olympics.coaches c
join tokyo_olympics.countries cn on c.country_id = cn.country_id
join tokyo_olympics.teams t on c.country_id = t.country_id and c.disc_id = t.disc_id
join tokyo_olympics.entries e on t.disc_id = e.disc_id
join tokyo_olympics.discipline d on d.disc_id = t.disc_id
where e.total_count > 10;

	coach_name	disc_name
1	GARNIER Valerie	Basketball
2	BERGE Christian	Handball
3	GORDO MANSILLA Juan David	Football
4	MIZURA Keiko	Hockey
5	MEZIANE Rachid	Basketball
6	NAIDOO Taren	Hockey
7	MATSUNO Yoshiyuki	Basketball
8	MAKIRE Hare	Rugby Sevens
9	SMITH Darren	Hockey
10	GRYZUNOVA Elena	Artistic Swimming
11	DIDENKO Natalia	Artistic Swimming
12	MARIJNE Sjoerd	Hockey
13	YURKIN Sergey	Volleyball
14	RODRIGUEZ Eduardo Rafael	Volleyball

12. What is the total number of medals each country won for disciplines with a specific gender participation?

```
select c.country_name, sum(m.total) as total_medals from tokyo_olympics.medals m
join tokyo_olympics.countries c on c.country_id = m.country_id
join tokyo_olympics.athletes a on c.country_id = a.country_id
join tokyo_olympics.entries e on a.disc_id = e.disc_id
where(e.female_count > 0 and e.male_count >0)
group by c.country_name
order by total_medals desc;
```

```

    | ✓
    | select c.country_name, sum(m.total) as total_medals from tokyo_olympics.medals m
    | join tokyo_olympics.countries c on c.country_id = m.country_id
    | join tokyo_olympics.athletes a on c.country_id = a.country_id
    | join tokyo_olympics.entries e on a.disc_id = e.disc_id
    | where(e.female_count > 0 and e.male_count >0)
    | group by c.country_name
    | order by total_medals desc;

```

The screenshot shows a database query editor with the following interface elements:

- Output Tab:** Shows the results of the executed query.
- Result 17:** Indicated the number of rows returned.
- Table Headers:** `country_name` and `total_medals`.
- Data Rows:** A list of 17 countries with their total medal counts, ordered by total medals in descending order.

country_name	total_medals
United States of America	68478
People's Republic of China	34144
Japan	33118
Great Britain	23725
ROC	21584
Australia	21022
Germany	14800
Italy	13640
France	12375
Netherlands	9828
Canada	8688
Brazil	6006
Spain	5372
Republic of Korea	4460

13. What are the names of Coaches who coached the Men's event for the United States of America?

```

select tc.coach_name, c.country_name, e.event_name, coalesce(d.disc_name,
'N/A') as discipline_name
from tokyo_olympics.coaches tc
join tokyo_olympics.countries c on c.country_id = tc.country_id
left outer join tokyo_olympics.discipline d on d.disc_id = tc.disc_id
join tokyo_olympics.events e on e.event_id = tc.event_id
where e.event_name = 'Men' and c.country_name = 'United States of America'
order by tc.coach_name;

```

```

✓ select tc.coach_name, c.country_name, e.event_name, coalesce(d.disc_name, 'N/A') as discipline_name
from tokyo_olympics.coaches tc
join tokyo_olympics.countries c on c.country_id = tc.country_id
left outer join tokyo_olympics.discipline d on d.disc_id = tc.disc_id
join tokyo_olympics.events e on e.event_id = tc.event_id
where e.event_name = 'Men' and c.country_name = 'United States of America'
order by tc.coach_name;

```

Output Result 18

coach_name	country_name	event_name	discipline_name
FRIDAY Mike	United States of America	Men	Rugby Sevens
FRIDAY Mike	United States of America	Men	Rugby Sevens
FRIDAY Mike	United States of America	Men	Rugby Sevens
FRIDAY Mike	United States of America	Men	Rugby Sevens
FRIDAY Mike	United States of America	Men	Rugby Sevens
FRIDAY Mike	United States of America	Men	Rugby Sevens
FRIDAY Mike	United States of America	Men	Rugby Sevens
FRIDAY Mike	United States of America	Men	Rugby Sevens
FRIDAY Mike	United States of America	Men	Rugby Sevens
POPOVICH Gregg	United States of America	Men	Basketball
POPOVICH Gregg	United States of America	Men	Basketball
POPOVICH Gregg	United States of America	Men	Basketball
POPOVICH Gregg	United States of America	Men	Basketball
POPOVICH Gregg	United States of America	Men	Basketball

14. What are the events and disciplines coached by each coach in the Tokyo 2021 Olympics? Let the event name be only events that have “ea” contained in it

```

select co.coach_name, e.event_name, d.disc_name from tokyo_olympics.coaches co
join tokyo_olympics.countries c on co.country_id = c.country_id
left join tokyo_olympics.events e on co.event_id = e.event_id
left join tokyo_olympics.discipline d on co.disc_id = d.disc_id
where event_name like '%ea%';

```

```

✓ select co.coach_name, e.event_name, d.disc_name from tokyo_olympics.coaches co
join tokyo_olympics.countries c on co.country_id = c.country_id
left join tokyo_olympics.events e on co.event_id = e.event_id
left join tokyo_olympics.discipline d on co.disc_id = d.disc_id
where event_name like '%ea%';

```

Output Result 19

coach_name	event_name	disc_name
MONTICO Loredana	Team	Artistic Swimming
SZAUDER Gabor	Team	Artistic Swimming
KULESZA Katarzyna	Team	Artistic Swimming
ELAFANDI Nour	Team	Artistic Swimming
CHEPAK Anastasiya	Team	Artistic Swimming
TARRES CAMPA Ana	Team	Artistic Swimming
CHERNETSKA Natalia	Team	Artistic Swimming
GIALLOMBARDO Patrizia	Team	Artistic Swimming
FARINELLI Roberta	Team	Artistic Swimming
WANG Jie	Team	Artistic Swimming
GRYUNOVA Elena	Team	Artistic Swimming
REQUENA PERICAS Judit	Team	Artistic Swimming
FUJIKI Mayuko	Team	Artistic Swimming
MEZHENINA Valeria	Team	Artistic Swimming

15. What countries have the most gold across all female events in the Tokyo 2021 Olympics?

```
select c.country_name, sum(m.gold) as total_gold_medals_female
from tokyo_olympics.countries c
    join tokyo_olympics.medals m on c.country_id = m.country_id
join tokyo_olympics.teams t on m.country_id = t.country_id
join tokyo_olympics.events e on e.event_id = t.event_id
join tokyo_olympics.entries en on e.disc_id = en.disc_id
where
    en.female_count > 0
group by c.country_name
order by total_gold_medals_female desc
```

```
| ✓ select c.country_name, sum(m.gold) as total_gold_medals_female
|   from tokyo_olympics.countries c
|     join tokyo_olympics.medals m on c.country_id = m.country_id
|       join tokyo_olympics.teams t on m.country_id = t.country_id
|         join tokyo_olympics.events e on e.event_id = t.event_id
|           join tokyo_olympics.entries en on e.disc_id = en.disc_id
|             where
|               en.female_count > 0
|             group by c.country_name
|             order by total_gold_medals_female desc;
```

Output Result 21 ×

65 rows

	country_name	total_gold_medals_female
1	Japan	1188
2	ROC	640
3	Great Britain	616
4	Australia	527
5	Italy	340
6	Germany	330
7	France	330
8	Netherlands	240
9	Canada	189
10	Brazil	147

16. List countries and their respective medal counts in Basketball events

```
select c.country_name, sum(m.gold + m.silver + m.bronze) as totl_medals
from tokyo_olympics.teams
join tokyo_olympics.countries c on c.country_id = teams.country_id
join tokyo_olympics.discipline d on d.disc_id = teams.disc_id
join tokyo_olympics.medals m on teams.country_id = m.country_id
where d.disc_name = 'Basketball'
group by c.country_name;
```

✓ select c.country_name, sum(m.gold + m.silver + m.bronze) as totl_medals
from tokyo_olympics.teams
join tokyo_olympics.countries c on c.country_id = teams.country_id
join tokyo_olympics.discipline d on d.disc_id = teams.disc_id
join tokyo_olympics.medals m on teams.country_id = m.country_id
where d.disc_name = 'Basketball'
group by c.country_name;

	country_name	totl_medals
1	Argentina	3
2	Australia	92
3	Belgium	7
4	Canada	24
5	Czech Republic	11
6	France	66
7	Germany	37

17. Who were the coaches for the Egyptian Football team in the Tokyo 2021 Olympics?

```
select distinct c.coach_name, c2.country_name, d.disc_name, e.event_name
from tokyo_olympics.coaches c
join tokyo_olympics.teams t on c.country_id = t.country_id
join tokyo_olympics.countries c2 on c2.country_id = t.country_id
join tokyo_olympics.discipline d on d.disc_id = t.disc_id
join tokyo_olympics.events e on e.event_id = t.event_id
where c2.country_name = 'Egypt' and d.disc_name = 'Football';
```

```

✓ select distinct c.coach_name, c2.country_name, d.disc_name,e.event_name
from tokyo_olympics.coaches c
join tokyo_olympics.teams t on c.country_id = t.country_id
join tokyo_olympics.countries c2 on c2.country_id = t.country_id
join tokyo_olympics.discipline d on d.disc_id = t.disc_id
join tokyo_olympics.events e on e.event_id = t.event_id
where c2.country_name = 'Egypt' and d.disc_name = 'Football';

```

	coach_name	country_name	disc_name	event_name
1	ABDELMAGID Wael	Egypt	Football	Men
2	ALY Kamal	Egypt	Football	Men
3	CHEPAK Anastasiya	Egypt	Football	Men
4	ELAFANDI Nour	Egypt	Football	Men
5	FAHMY LATIF Yasmine	Egypt	Football	Men
6	MOHAMED Mai	Egypt	Football	Men
7	PARRONDO Roberto	Egypt	Football	Men
8	SALAM Mohamed	Egypt	Football	Men
9	SHAWKY Gharib	Egypt	Football	Men
10	SIAM Zaghloul	Egypt	Football	Men

18. How many events are allocated to each discipline and country, considering only those disciplines where a country has athletes?

```

SELECT d.disc_name, c.country_name, COUNT(DISTINCT e.event_id) AS num_events
FROM tokyo_olympics.Discipline d
JOIN tokyo_olympics.Teams t ON d.disc_id = t.disc_id
JOIN tokyo_olympics.Events e ON t.event_id = e.event_id
JOIN tokyo_olympics.Countries c ON t.country_id = c.country_id
GROUP BY d.disc_name, c.country_name;

```

```

✓ SELECT d.disc_name, c.country_name, COUNT(DISTINCT e.event_id) AS num_events
FROM tokyo_olympics.Discipline d
JOIN tokyo_olympics.Teams t ON d.disc_id = t.disc_id
JOIN tokyo_olympics.Events e ON t.event_id = e.event_id
JOIN tokyo_olympics.Countries c ON t.country_id = c.country_id
GROUP BY d.disc_name, c.country_name;

```

	disc_name	country_name	num_events
1	3x3 Basketball	Belgium	1
2	3x3 Basketball	France	1
3	3x3 Basketball	Italy	1
4	3x3 Basketball	Japan	2
5	3x3 Basketball	Latvia	1
6	3x3 Basketball	Mongolia	1
7	3x3 Basketball	Netherlands	1
8	3x3 Basketball	Poland	1
9	3x3 Basketball	ROC	2
10	3x3 Basketball	Romania	1
11	3x3 Basketball	Serbia	1
12	Archery	Australia	2

19. What are the names of coaches who have athletes participating in both Swimming and Rhythmic Gymnastics? Also include the name of the countries.

```
select c.coach_name, co.country_name
from tokyo_olympics.coaches c
join tokyo_olympics.countries co on co.country_id = c.country_id
join tokyo_olympics.athletes a on co.country_id = a.country_id
join tokyo_olympics.discipline d on d.disc_id = a.disc_id
where d.disc_name in ('Swimming', 'Rhythmic Gymnastics')
group by c.coach_name, co.country_name
having count(distinct d.disc_id) = 2;
```

✓ select c.coach_name, co.country_name
from tokyo_olympics.coaches c
join tokyo_olympics.countries co on co.country_id = c.country_id
join tokyo_olympics.athletes a on co.country_id = a.country_id
join tokyo_olympics.discipline d on d.disc_id = a.disc_id
where d.disc_name in ('Swimming', 'Rhythmic Gymnastics')
group by c.coach_name, co.country_name
having count(distinct d.disc_id) = 2;

Output Result 25 ×

	coach_name	country_name
1	ABDELMAGID Wael	Egypt
2	ABE Junya	Japan
3	ABE Katsuhiko	Japan
4	AGEBA Yuya	Japan
5	AIKMAN Siegfried Gottlieb	Japan
6	ALEKSEEV Alexey	ROC
7	ALY Kamal	Egypt
8	ANDONOVSKI Vlatko	United States of America
9	ARNAU CREUS Xavier	Japan
10	ARNOLD Graham	Australia

20. Display events together with corresponding discipline, coach and country?

```
select e.event_name, d.disc_name, coalesce(c.coach_name, 'No coach assigned')  
as coach_name, cn.country_name  
from tokyo_olympics.events e  
join tokyo_olympics.discipline d on d.disc_id = e.disc_id  
left join tokyo_olympics.coaches c on e.event_id = c.event_id and e.disc_id =  
c.disc_id  
left outer join tokyo_olympics.countries cn on cn.country_id = c.country_id;
```

The screenshot shows a database query interface. At the top, there is a code editor containing the SQL query. Below the code editor is a results table. The results table has four columns: event_name, disc_name, coach_name, and country_name. The data in the table is as follows:

	event_name	disc_name	coach_name	country_name
1	Team	Artistic Swimming	MONTICO Loredana	Australia
2	Team	Artistic Swimming	SZAUDER Gabor	Canada
3	Team	Artistic Swimming	KULESZA Katarzyna	Canada
4	Team	Artistic Swimming	ELAFANDI Nour	Egypt
5	Team	Artistic Swimming	CHEPAK Anastasiya	Egypt
6	Team	Artistic Swimming	TARRES CAMPA Ana	Greece
7	Team	Artistic Swimming	CHERNETSKA Natalia	Greece
8	Team	Artistic Swimming	GIALLOMBARDO Patrizia	Italy
9	Team	Artistic Swimming	FARINELLI Roberta	Italy
10	Team	Artistic Swimming	WANG Jie	People's Republic of China
11	Team	Artistic Swimming	GRYZUNOVA Elena	ROC
12	Team	Artistic Swimming	REQUENA PERICAS Judit	Spain