Commonwealth Games Management Database

Team ID: T607

Members:

1) Harshvardhan Vajani (202201413)
Contact No. :- 9558156560

2) Maahi Shah (202201419)

3) Heet Thakkar (202201431)

4) Abhishek Abbi (202201438)

5) Parth Tolani (202201455)

Made Under the guidance of Sneh Raval

Summary

a) While working on this project, we'd gained some valuable insights regarding the approach for making any kind of Database. We came across some of the challenges, which were not so major, but hindered, thereby deviating from our thought about the project. The major challenge was in the making of the correct **ER-Diagram**, we updated it whenever we felt the need to do so and tried our level best to make it close to the original idea we had started with in our minds. Apart from it, insertion was bit of a tedious task, however not a challenge, but since we changed ER quite a few times, it became a bit more tedious. However, we managed to pull it off as a team.

b) We have tried to incorporate some real-world instances into our project and to make it more realistic, we often visited the official website of the Commonwealth Games, tried to make some of the portions better than it.

c) We'd all had a great add-on from this project, for it enhanced our knowledge of DBMS from theory, to real-world like simulation, as to how some concepts get used in real life. While in the making of the project, we realized there was more to it, bit challenging, tedious, however, it was so much fun to get to learn and incorporate things from the Lectures in the project.

This project proposes a revolutionary approach to managing the Commonwealth Games through a comprehensive database system. This system aims to streamline operations, enhance user experience, and elevate the overall event experience.

Introduction

Current Commonwealth Games management likely involves multiple systems or manual processes, leading to inefficiencies. This project proposes a centralized database system that acts as a single source of truth for all Games-related data.

The system caters to a diverse user base:

- **Public:** Easy access to real-time schedules, results, and general information.
- **Athletes:** Seamless registration, personalized schedules, and relevant updates.
- **Organizing Committee:** Efficient data management for informed decision-making.
- **Venue Managers:** Real-time information for optimal venue operations.

Objectives

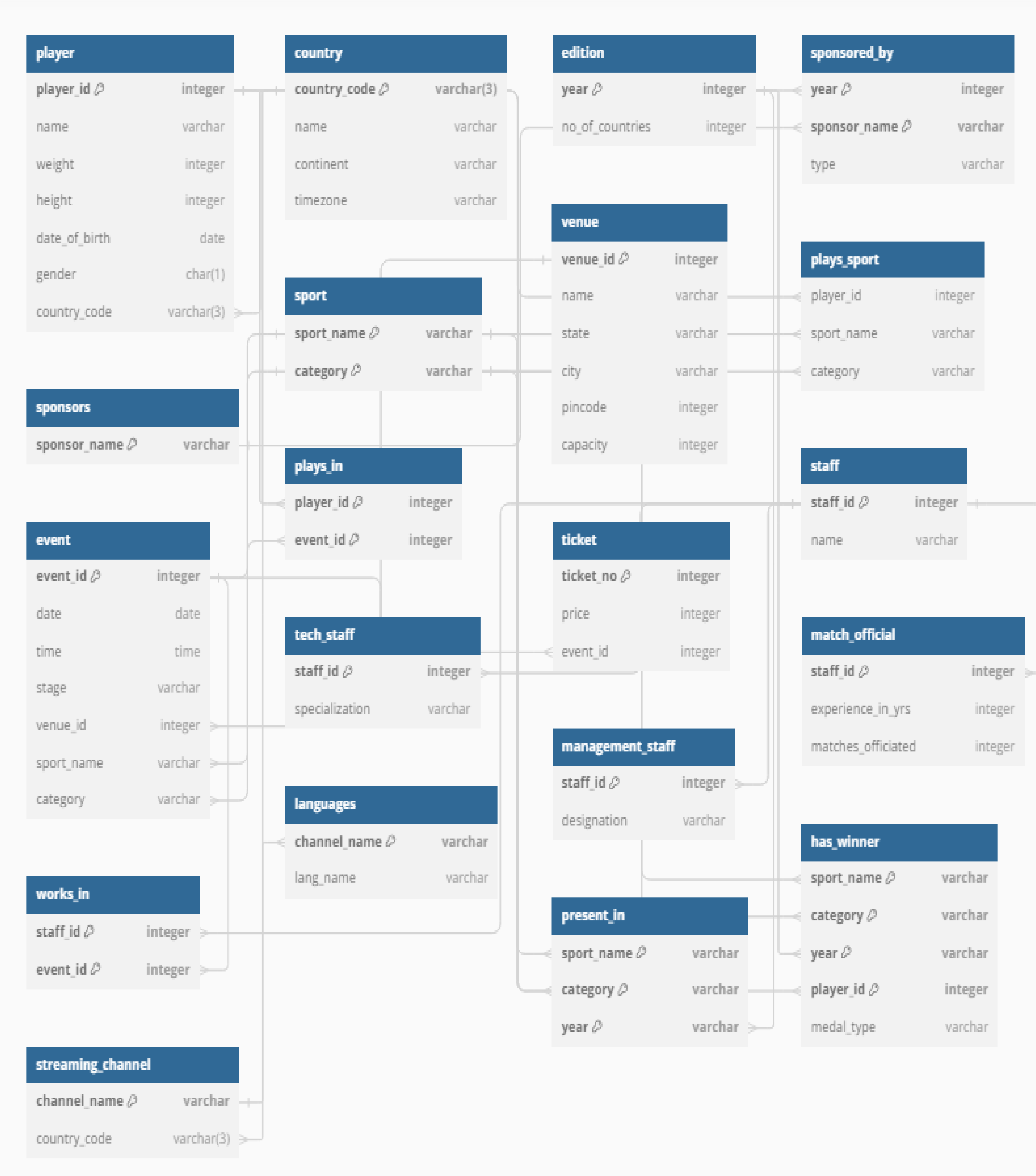
The core objective is to design and implement a robust database system that manages all critical aspects of the Commonwealth Games, including:

- **Athlete Registration:** Secure and efficient registration process for athletes.
- **Event Scheduling:** Comprehensive scheduling with real-time updates and conflict resolution.
- **Ticketing:** Secure online ticketing platform with seat selection and management functionalities.
- **Venue Management:** Real-time venue information, resource allocation, and operational tools.
- **Result Tracking:** Efficient system for capturing, storing, and displaying competition results.

Expected Outcomes:

- **Enhanced Efficiency:** Streamlined operations through a centralized data platform.
- **Improved User Experience:** Seamless interaction and real-time information access for all users.
- **Informed Decision-Making:** Data-driven insights for better planning and resource allocation.
- **Elevated Event Experience:** Increased transparency and accessibility for athletes, spectators, and organizers.This project presents a significant advancement in Commonwealth Games management, fostering a more efficient, organized, and user-centric experience for athletes, organizers, and the public.

RELATIONAL SCHEMA



player	
player_id 🔑	integer
name	varchar
weight	integer
height	integer
date_of_birth	date
gender	char(1)
country_code	varchar(3)

country	
country_code 🔑	varchar(3)
name	varchar
continent	varchar
timezone	varchar

edition	
year 🔑	integer
no_of_countries	integer

sponsored_by	
year 🔑	integer
sponsor_name 🔑	varchar
type	varchar

venue	
venue_id 🔑	integer
name	varchar
state	varchar
city	varchar
pincode	integer
capacity	integer

sport	
sport_name 🔑	varchar
category 🔑	varchar

plays_sport	
player_id	integer
sport_name	varchar
category	varchar

sponsors	
sponsor_name 🔑	varchar

plays_in	
player_id 🔑	integer
event_id 🔑	integer

staff	
staff_id 🔑	integer
name	varchar

event	
event_id 🔑	integer
date	date
time	time
stage	varchar
venue_id	integer
sport_name	varchar
category	varchar

ticket	
ticket_no 🔑	integer
price	integer
event_id	integer

tech_staff	
staff_id 🔑	integer
specialization	varchar

match_official	
staff_id 🔑	integer
experience_in_yrs	integer
matches_officiated	integer

management_staff	
staff_id 🔑	integer
designation	varchar

languages	
channel_name 🔑	varchar
lang_name	varchar

has_winner	
sport_name 🔑	varchar
category 🔑	varchar
year 🔑	varchar
player_id 🔑	integer
medal_type	varchar

works_in	
staff_id 🔑	integer
event_id 🔑	integer

present_in	
sport_name 🔑	varchar
category 🔑	varchar
year 🔑	varchar

streaming_channel	
channel_name 🔑	varchar
country_code	varchar(3)

BCNF CONVERSION

TABLE	FUNCTIONAL DEPENDENCIES	BCNF	CONVERSION TO BCNF
Player	player_id -> player_name player_id -> weight player_id -> height player_id -> date_of_birth player_id -> gender player_id -> country_code	☑	
Country	country_code -> country_name country_code -> continent country_code -> timezone country_name -> country_code country_name -> continent country_name -> timezone	☑	
Edition	year -> no_of_countries	☑	
Sponsored_by	{sponsor_name,year} -> type	☑	
Sport	{sport_name,category} -> {sport_name,category}	☑	
Venue	venue_id -> venue_name venue_id -> state venue_id -> city venue_id -> capacity venue_id -> pincode pincode -> city pincode -> state city -> state	☒	Cannot be converted to BCNF as it consists of composite keys.
Event	event_id -> date event_id -> time event_id -> stage event_id -> venue_id event_id -> sport_name event_id -> category	☑	
Sponsors	sponsor_name -> sponsor_name	☑	
Ticket	ticket_no -> price ticket_no -> event_id	☑	

Plays_in	{player_id,event_id} -> {player_id,event_id}	☑	
Streaming_channel	{channel_name,country_code} -> {channel_name,country_code}	☑	
Languages	{channel_name,lang_name} -> {channel_name,lang_name}	☑	
Present_in	{sport_name,category,year} -> {sport_name,category,year}	☑	
Has_winner	{player_id,category,sport_name,year} -> medal_type	☑	
Staff	staff_id -> name	☑	
Tech_staff	staff_id -> specialization	☑	
Management_staff	staff_id -> designation	☑	
Match_official	staff_id -> experience_in_years staff_id -> matches_officiated	☑	
Works_in	{staff_id,event_id} -> {staff_id,event_id}	☑	
Plays_sport	{player_id,sport_name,category} -> {player_id,sport_name,category}	☑	

Proofs and Justifications

In BCNF

The tables player, country, edition, sponsored_by, sport, event, sponsors, plays_in, streaming_channel, languages, present_in, has_winner, staff, tech_staff, management_staff, match_official, Ticket and Works_in are all in BCNF. This is because their primary keys are super keys, meaning that they uniquely determine all other attributes in the table.

NOT in BCNF

The table Venue is not in BCNF.
In the Venue table, while venue_id serves as a super key by uniquely determining all other attributes, pincode only functionally determines city and state, and city only functionally determines state, neither of which qualifies as a super key.

DDL

```
CREATE TABLE player (
    player_id INTEGER PRIMARY KEY,
    name VARCHAR,
    weight INTEGER,
    height INTEGER,
    date_of_birth DATE,
    gender CHAR(1),
    country_code VARCHAR(3)
 );

 CREATE TABLE country (
    country_code VARCHAR(3) PRIMARY KEY,
    name VARCHAR,
    continent VARCHAR,
    timezone VARCHAR
 );

 CREATE TABLE edition (
    year INTEGER PRIMARY KEY,
    no_of_countries INTEGER
 );

 CREATE TABLE sponsored_by (
    year INTEGER,
    sponsor_name VARCHAR,
    sponsor_type VARCHAR,
    PRIMARY KEY (year, sponsor_name)
 );


 CREATE TABLE sponsors (
    sponsor_name VARCHAR PRIMARY KEY
 );

 CREATE TABLE sport (
    sport_name VARCHAR,
    category VARCHAR,
    PRIMARY KEY (sport_name, category)
 );
```

```
CREATE TABLE venue (
   venue_id INTEGER PRIMARY KEY,
   name VARCHAR,
   state VARCHAR,
   city VARCHAR,
   pincode INTEGER,
   capacity INTEGER
);

CREATE TABLE plays_sport (
   player_id INTEGER,
   sport_name VARCHAR,
   category VARCHAR,
   PRIMARY KEY (player_id, sport_name, category),
   FOREIGN KEY (player_id) REFERENCES player(player_id),
   FOREIGN KEY (sport_name, category) REFERENCES sport(sport_name, category)
);

CREATE TABLE event (
   event_id INTEGER PRIMARY KEY,
   date DATE,
   time TIME,
   stage VARCHAR,
   venue_id INTEGER,
   sport_name VARCHAR,
   category VARCHAR,
   FOREIGN KEY (venue_id) REFERENCES venue(venue_id),
   FOREIGN KEY (sport_name, category) REFERENCES sport(sport_name, category)
);

CREATE TABLE plays_in (
   player_id INTEGER,
   event_id INTEGER,
   PRIMARY KEY (player_id, event_id),
   FOREIGN KEY (player_id) REFERENCES player(player_id),
   FOREIGN KEY (event_id) REFERENCES event(event_id)
);

CREATE TABLE ticket (
   ticket_no INTEGER PRIMARY KEY,
   price INTEGER,
   event_id INTEGER,

   FOREIGN KEY (event_id) REFERENCES event(event_id),
);

CREATE TABLE staff (
    staff_id INTEGER PRIMARY KEY,
    name VARCHAR
);

CREATE TABLE works_in (
    staff_id INTEGER,
    event_id INTEGER,

    PRIMARY KEY (staff_id, event_id),
    FOREIGN KEY (staff_id) REFERENCES staff(staff_id),
    FOREIGN KEY (event_id) REFERENCES event(event_id)

);

CREATE TABLE tech_staff (
    staff_id INTEGER PRIMARY KEY,
    specialization VARCHAR,
    FOREIGN KEY (staff_id) REFERENCES staff(staff_id)
);

CREATE TABLE management_staff (
    staff_id INTEGER PRIMARY KEY,
    designation VARCHAR,
    FOREIGN KEY (staff_id) REFERENCES staff(staff_id)
);

CREATE TABLE match_official (
    staff_id INTEGER PRIMARY KEY,
    experience_in_yrs INTEGER,
    matches_officiated INTEGER,
    FOREIGN KEY (staff_id) REFERENCES staff(staff_id)
);

CREATE TABLE streaming_channel (
    channel_name VARCHAR PRIMARY KEY,
    country_code VARCHAR(3),
    FOREIGN KEY (country_code) REFERENCES country(country_code)
);
```

```
CREATE TABLE languages (
    channel_name VARCHAR,
    lang_name VARCHAR,
    PRIMARY KEY (channel_name, lang_name),
    FOREIGN KEY (channel_name) REFERENCES streaming_channel(channel_name)
);

CREATE TABLE present_in (
    sport_name VARCHAR,
    category VARCHAR,
    year INTEGER,
    PRIMARY KEY (sport_name, category, year),
    FOREIGN KEY (sport_name, category) REFERENCES sport(sport_name, category),
    FOREIGN KEY (year) REFERENCES edition(year)
);

CREATE TABLE has_winner (
    sport_name VARCHAR,
    category VARCHAR,
    year INTEGER,
    player_id INTEGER,
    medal_type VARCHAR,
    PRIMARY KEY (sport_name, category, year, player_id),
    FOREIGN KEY (sport_name, category) REFERENCES sport(sport_name, category),
    FOREIGN KEY (year) REFERENCES edition(year),
    FOREIGN KEY (player_id) REFERENCES player(player_id)
);
```

Queries

```
-- 1) Find the details of all events that are yet to happen (future events):

SELECT event_id, date, time, stage
FROM event
WHERE date > '2020-08-12';

-- 2) Retrieve the total number of medals won by each country in a specific year:

SELECT p.country_code, COUNT(hw.medal_type) AS total_medals
FROM player p
JOIN has_winner AS hw ON p.player_id = hw.player_id
WHERE hw.year = 2020
GROUP BY p.country_code;


--3) Find the top 3 most popular sports based on the number of ticket sold :

SELECT  event.sport_name,event.category, COUNT(*) AS total_tickets_sold
FROM ticket
JOIN event ON ticket.event_id = event.event_id
GROUP BY event.event_id, event.sport_name
ORDER BY total_tickets_sold DESC
LIMIT 3;


--4) List all sponsors for a specific year 2020:

SELECT sponsor_name
FROM sponsored_by
WHERE year = 2020;

--5) Retrieve all players who have won medals in a 2020:

 SELECT player.name, has_winner.medal_type
FROM player
```

```sql
JOIN has_winner ON player.player_id = has_winner.player_id
WHERE has_winner.year = 2020;


-- 6)Count the number of tickets sold for an event with event_id 5:

SELECT COUNT(ticket_no) AS total_tickets_sold
FROM ticket
WHERE event_id = 5;


-- 7)Find all staff members working in a particular venue with venue_id 16 during a specific event with event_id 5:

SELECT staff.name
FROM staff
JOIN works_in ON staff.staff_id = works_in.staff_id
WHERE works_in.event_id = 5;

-- 8) List all streaming channels available in a specific country with country code IND

SELECT channel_name
FROM streaming_channel
WHERE country_code = 'IND';


--9) Retrieve all events where a specific player with player_id 13  is participating:

SELECT e.event_id, date, time, stage
FROM event as e
JOIN plays_in ON e.event_id = plays_in.event_id
WHERE plays_in.player_id = 13;


-- 10) Count the number of matches officiated by each match official:

SELECT staff_id, (matches_officiated) AS total_matches_officiated
FROM match_official
GROUP BY staff_id;
```

```sql
-- 11. Retrieve the names of players who are playing in multiple sports:

SELECT DISTINCT player.name3
FROM player
JOIN plays_sport ON player.player_id = plays_sport.player_id
GROUP BY player.player_id
HAVING COUNT(DISTINCT plays_sport.sport_name) > 1;

-- 12. Find all events scheduled on a 2020-08-05:

SELECT event_id, date, time, stage
FROM event
WHERE date = '2020-08-05';


-- 13. List all venues and their capacities in a particular city:

SELECT name, capacity
FROM venue
WHERE city = 'Ahmedabad';

-- 14. Retrieve the names of all players who have won a gold medal:

SELECT player.name,has_winner.sport_name, has_winner.category
FROM player
JOIN has_winner ON player.player_id = has_winner.player_id
WHERE has_winner.medal_type = 'gold';


-- 15. Count the number of events in each sport category:

SELECT sport_name, category, COUNT(*) AS total_events
FROM event
GROUP BY sport_name, category;

-- 16. Find the average ticket price for events held in a specific venue:

SELECT AVG(price) AS average_ticket_price
FROM ticket
WHERE event_id = 9;
```

```
-- 17. Count the total number of events in each sport category held in a specific year:


SELECT s.category, COUNT(*) AS total_events
FROM event e
JOIN sport AS s ON e.sport_name = s.sport_name
WHERE YEAR(e.date) = 2020
GROUP BY s.category;


-- 18. List all players from a specific country participating in the games:

SELECT name
FROM player
WHERE country_code = 'CAN';


-- 19. List the name of all staff members who are specialized in a particular technology:

SELECT s.name
FROM staff s
JOIN tech_staff ts ON s.staff_id = ts.staff_id
WHERE ts.specialization = 'Web Development';

-- 20. List all players participating in male-T20 cricket:

SELECT player.name
FROM player
JOIN plays_sport ON player.player_id = plays_sport.player_id
JOIN sport ON plays_sport.sport_name = sport.sport_name
WHERE sport.sport_name = 'Cricket'
AND sport.category = 'male-T20';

-- 21.  Retrieve the total number of countries participating in each edition of the games:

SELECT year, no_of_countries
FROM edition;
```

-- 22. Find all details of events scheduled in a particular venue 7:

```
SELECT event_id, date, time, stage, venue.name
FROM event
JOIN venue ON event.venue_id = venue.venue_id
WHERE venue.venue_id = 7;
```

--23) List all streaming channels available for a specific language:

```
SELECT sc.channel_name
FROM streaming_channel sc
JOIN languages l ON sc.channel_name = l.channel_name
WHERE l.lang_name = 'English';
```

-- 24. Retrieve the names of all players who have won a medal in a particular sport:

```
SELECT DISTINCT p.name, hw.category
FROM player as  p
JOIN has_winner as  hw ON p.player_id = hw.player_id
JOIN event e ON (hw.sport_name = e.sport_name AND hw.category= e.category)
WHERE e.sport_name = 'Race'
AND hw.medal_type IS NOT NULL;
```

-- 25. List all players who have won medals in multiple events:

```
SELECT
   hw.sport_name,
   hw.category,
   hw.medal_type,
   p.name AS player_name
FROM
   has_winner hw
JOIN
   player p ON hw.player_id = p.player_id
WHERE
   hw.player_id IN (
      SELECT player_id
      FROM has_winner
      GROUP BY player_id
```

```
        HAVING COUNT(*) > 1
    );


-- 26. Retrieve the names of all sports categories available in the system:

SELECT DISTINCT category,sport_name
FROM sport;


--27. Count the total number of events held in each venue:

SELECT venue.name, COUNT(*) AS total_events
FROM event
JOIN venue ON event.venue_id = venue.venue_id
GROUP BY venue.name;

-- 28. Find the total revenue generated from ticket sales for all events in a specific year:

SELECT SUM(price) AS total_revenue
FROM ticket
JOIN event ON ticket.event_id = event.event_id
WHERE extract(YEAR from event.date) = 2020;

--29. Count the total number of medals won by each player in a specific year:

SELECT p.name, COUNT(hw.medal_type) AS total_medals
FROM player p
LEFT JOIN has_winner hw ON p.player_id = hw.player_id
WHERE hw.year = 2020
GROUP BY p.name;

--30 List all events scheduled on a specific day of the week:

SELECT event_id, date, time, stage
FROM event
WHERE EXTRACT(DOW FROM date) = 6;
```

```
--31 List all events scheduled in a specific year and sorted by date and time:

SELECT event_id, date, time, stage
FROM event
WHERE YEAR(date) = 2020
ORDER BY date, time;
```

```
-- 32 Find the top 3 countries with the highest number of gold medals in a specific year:


SELECT p.country_code, COUNT(hw.medal_type) AS gold_medals
FROM player p
JOIN has_winner hw ON p.player_id = hw.player_id
WHERE hw.medal_type = 'gold' AND hw.year = 2020
GROUP BY p.country_code
ORDER BY gold_medals DESC
LIMIT 3;
```