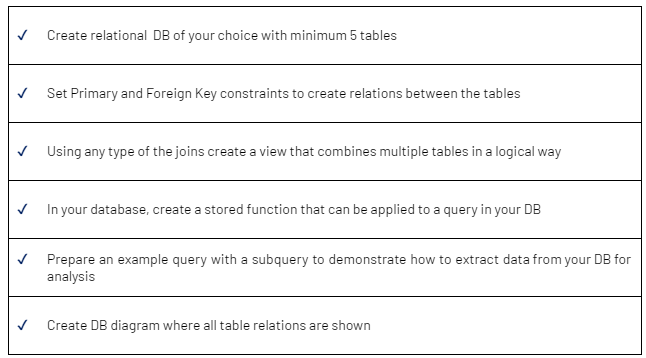
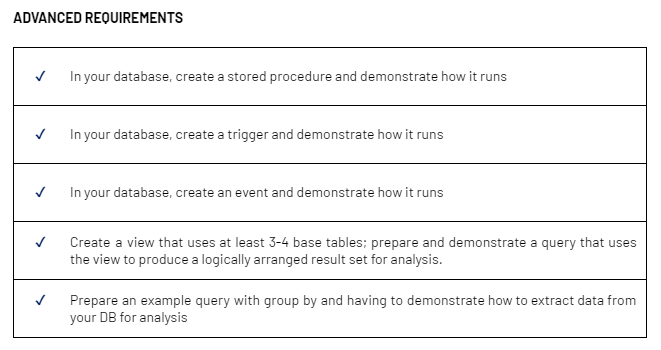
* 



* **Tables required to organise the Hunger Games:**

Districts:

* + District\_ID (Primary Key)
  + District\_Name
  + District\_Descriptor
* Tributes:
  + Tribute\_ID (Primary Key)
  + FirstName
  + LastName
  + District\_ID (Foreign Key referencing Districts)

Age

* + Tribute\_ID (Foreign Key referencing Tributes)
  + Age
  + Gender
  + Skills
* Event\_Name:
  + Event\_ID (Primary Key)
  + Event\_Type (e.g., Cornucopia Bloodbath, Tracker Jacker Attack, etc.)
  + Description
  + EventDateTime
* Sponsors:
  + Sponsor\_ID (Primary Key)
  + SponsorName
  + Tribute\_ID (Foreign Key referencing Tributes)
  + AmountDonated
* Kills:
  + Kill\_ID (Primary Key)
  + Killer\_ID (Foreign Key referencing Tributes)
  + Victim\_ID (Foreign Key referencing Tributes)
  + KillDateTime
  + Location (foreign key referencing Arena\_ID)
* Arenas:
  + Arena\_ID (Primary Key)
  + ArenaName
  + Description
  + Features (terrain, obstacles, etc.)

Weapons:

* Weapon\_ID (Primary Key)
* Weapon\_Type

TributeWeapons:

* Tribute\_ID (Foreign Key referencing Tributes)
* Weapon\_ID(Foreign Key referencing Weapons)

Mentors:

* Mentor\_ID (Primary Key)
* FirstName
* LastName
* District\_ID (Foreign Key referencing Districts)

**Code to create Hunger Games Database:**

CREATE DATABASE HungerGames;

USE HungerGames;

CREATE TABLE Districts (

District\_ID INT NOT NULL,

District\_Name VARCHAR(20) NOT NULL,

District\_Descriptor VARCHAR(20) NOT NULL,

PRIMARY KEY(District\_ID)

);

CREATE TABLE Weapons (

Weapon\_ID INT NOT NULL,

Weapon\_Type VARCHAR(50) NOT NULL,

PRIMARY KEY(Weapon\_ID)

);

CREATE TABLE Tributes (

Tribute\_ID INT,

FirstName VARCHAR(20) NOT NULL,

LastName VARCHAR(20) NOT NULL,

District\_ID INT NOT NULL,

PRIMARY KEY (Tribute\_ID),

CONSTRAINT FK\_DISTRICT

FOREIGN KEY(District\_ID)

REFERENCES Districts(District\_ID)

);

CREATE TABLE Age (

Tribute\_ID INT,

Age INT NOT NULL,

Gender VARCHAR(2) NOT NULL,

Skills VARCHAR(20) NOT NULL,

CONSTRAINT FK\_ID\_TRIBUTE

FOREIGN KEY(Tribute\_ID)

REFERENCES Tributes(Tribute\_ID)

);

CREATE TABLE Tribute\_Weapons (

Tribute\_ID INT NOT NULL,

Weapon\_ID INT NOT NULL,

CONSTRAINT FK\_Weapon\_Tribute

FOREIGN KEY (Tribute\_ID)

REFERENCES Tributes(Tribute\_ID),

CONSTRAINT FK\_Weapon\_ID

FOREIGN KEY (Weapon\_ID)

REFERENCES Weapons(Weapon\_ID)

);

CREATE TABLE Event\_Name (

Event\_ID INT NOT NULL,

Event\_Type VARCHAR(50) NOT NULL,

Description VARCHAR(150) NOT NULL,

EventDateTime datetime,

PRIMARY KEY(Event\_ID)

);

CREATE TABLE Arenas (

Arena\_ID INT NOT NULL,

ArenaName VARCHAR(20) NOT NULL,

Description VARCHAR(150) NOT NULL,

Features VARCHAR(50),

PRIMARY KEY (Arena\_ID)

);

CREATE TABLE Sponsors (

Sponsor\_ID INT NOT NULL,

SponsorName VARCHAR (20),

Tribute\_ID INT NOT NULL,

AmountDonated DECIMAL (12, 2),

PRIMARY KEY(Sponsor\_ID),

CONSTRAINT FK\_Tribute\_ID

FOREIGN KEY(Tribute\_ID)

REFERENCES Tributes(Tribute\_ID)

);

CREATE TABLE Kills (

Kill\_ID INT NOT NULL,

Killer\_ID INT,

Victim\_ID INT,

KillDateTime DATETIME,

Location INT,

PRIMARY KEY(Kill\_ID),

CONSTRAINT FK\_Killer\_ID

FOREIGN KEY(Killer\_ID)

REFERENCES Tributes(Tribute\_ID),

CONSTRAINT FK\_Victim\_ID

FOREIGN KEY(Victim\_ID)

REFERENCES Tributes(Tribute\_ID),

CONSTRAINT FK\_Location

FOREIGN KEY(Location)

REFERENCES Arenas(Arena\_ID)

);

CREATE TABLE Mentors (

Mentor\_ID INT NOT NULL,

FirstName VARCHAR(50) NOT NULL,

LastName VARCHAR(50) NOT NULL,

District\_ID INT NOT NULL,

PRIMARY KEY (Mentor\_ID),

Constraint FK\_District\_ID

FOREIGN KEY (District\_ID)

REFERENCES Districts(District\_ID)

);

**Data Info:**

USE HungerGames;

INSERT INTO Districts (District\_ID, District\_Name, District\_Descriptor)

VALUES

(1, 'District 1', 'Luxury'),

(2, 'District 2', 'Masonry'),

(3, 'District 3', 'Technology'),

(4, 'District 4', 'Fishing'),

(5, 'District 5', 'Power'),

(6, 'District 6', 'Transportation'),

(7, 'District 7', 'Lumber'),

(8, 'District 8', 'Textiles'),

(9, 'District 9', 'Grain'),

(10, 'District 10', 'Livestock'),

(11, 'District 11', 'Agriculture'),

(12, 'District 12', 'Mining');

INSERT INTO Tributes (Tribute\_ID, FirstName, LastName, District\_ID)

VALUES

(1, 'Katniss', 'Everdeen', 12),

(2, 'Peeta', 'Mellark', 12),

(3, 'Cato', 'Hadley', 2),

(4, 'Rue', 'Lorin', 11),

(5, 'Finnick', 'Odair', 4),

(6, 'Johanna', 'Mason', 7),

(7, 'Glimmer', 'Summers', 1),

(8, 'Thresh', 'Sonner', 11),

(9, 'Beetee', 'Latier', 3),

(10, 'Annie', 'Cresta', 4),

(11, 'Lester','Watt',5),

(12,'Amber', 'Fiate', 6),

(13,'Odin','Cottons', 8),

(14, 'Felicity','Jower', 9),

(15,'Dream','Lambe',10);

INSERT INTO Age (Tribute\_ID, Age, Gender, Skills)

VALUES

(1, 17, 'F', 'Archery'),

(2, 18, 'M', 'Baking'),

(3, 18, 'M', 'Swordsmanship'),

(4, 12, 'F', 'Botany'),

(5, 24, 'M', 'Trident proficiency'),

(6, 22, 'F', 'Axe throwing'),

(7, 21, 'F', 'Throwing knives'),

(8, 20, 'M', 'Strength'),

(9, 32, 'M', 'Electronics'),

(10, 23, 'F', 'Swimming'),

(11, 16, 'M', 'Electronics'),

(12, 15, 'F', 'Metal Working'),

(13, 17, 'M', 'Axe Throwing'),

(14, 13, 'F', 'Botany'),

(15, 18, 'F', 'Camoflage');

INSERT INTO Event\_Name (Event\_ID, Event\_Type, Description, EventDateTime)

VALUES

('1', 'Opening Ceremony', 'Grand opening of the Hunger Games', '2022-05-01 12:00:00'),

('2', 'Bloodbath', 'Initial fight for resources', '2022-05-02 09:00:00'),

('3', 'Alliance Formation', 'Tributes form alliances for survival', '2022-05-03 14:30:00'),

('4', 'Mutant Encounter', 'Tributes face off against mutant creatures', '2022-05-04 11:45:00'),

('5', 'Sponsor Gifts', 'Capitol sponsors send gifts to tributes', '2022-05-05 18:00:00'),

('6', 'Aerial Assault', 'Capitol-controlled drones attack tributes', '2022-05-06 16:30:00'),

('7', 'Tribute Interviews', 'Tributes showcase their skills and stories', '2022-05-07 10:00:00'),

('8', 'Midnight Feast', 'A feast with essential supplies announced at midnight', '2022-05-08 00:00:00'),

('9', 'Tribute Revolt', 'Tributes rebel against Capitol control', '2022-05-09 13:15:00'),

('10', 'Final Showdown', 'The last tributes face off for victory', '2022-05-10 17:45:00');

INSERT INTO Sponsors (Sponsor\_ID, SponsorName, Tribute\_ID, AmountDonated)

VALUES

(1, 'Capitol Sponsor 1', 1, 500.00),

(2, 'District 2 Sponsor', 3, 300.00),

(3, 'Capitol Sponsor 2', 5, 700.00),

(4, 'District 11 Sponsor', 8, 250.00),

(5, 'Capitol Sponsor 3', 10, 450.00),

(6, 'District 12 Sponsor', 12, 200.00),

(7, 'Capitol Sponsor 4', 2, 600.00),

(8, 'District 4 Sponsor', 4, 350.00),

(9, 'Capitol Sponsor 5', 6, 400.00),

(10, 'District 7 Sponsor', 7, 320.00),

(11, 'District 9 Sponsor', 14, 100.00),

(12, 'Capitol Sponsor 6', 1, 320.00);

INSERT INTO Arenas (Arena\_ID, ArenaName, Description, Features)

VALUES

(1, 'Forest Arena', 'A dense forest with hidden dangers', 'Mutant creatures, traps'),

(2, 'Mountain Arena', 'Treacherous mountain terrain', 'Rockslides, thin air'),

(3, 'Desert Arena', 'Endless sand dunes under scorching sun', 'Sandstorms, oasis'),

(4, 'Frozen Tundra', 'A frigid landscape with icy winds', 'Frozen lakes, ice caves'),

(5, 'Urban Jungle', 'Abandoned city with overgrown vegetation', 'Collapsed buildings, hidden passages'),

(6, 'Volcanic Crater', 'A smoky and dangerous volcanic crater', 'Lava pits, volcanic eruptions'),

(7, 'Underground Bunker', 'A network of dark and claustrophobic tunnels', 'Booby traps, limited visibility'),

(8, 'Island Paradise', 'Tropical island surrounded by treacherous waters', 'Sharks, coral reefs'),

(9, 'Sky High Arena', 'Floating islands in the sky', 'Floating platforms, unpredictable weather'),

(10, 'Maze of Mirrors', 'A confusing labyrinth of mirrors', 'Illusions, disorientation');

INSERT INTO Kills (Kill\_ID, Killer\_ID, Victim\_ID, KillDateTime,Location)

VALUES

(1, 3, 15, '2022-05-02 09:01:00',1), -- Cato kills Dream

(2, 7, 12, '2022-05-02 09:05:00',1), -- Glimmer kills Amber

(3, 14, 11, '2022-05-02 09:27:00',1), -- Felicity kills Lester

(4, 13, 10, '2022-05-03 10:30:00',3), -- Odin kills Annie

(5, 1, 6, '2022-05-04 11:50:00',2), -- Cato Kills Joanna

(6, 8, 7, '2022-05-04 12:05:00',4), -- Thresh Kills Glimmer

(7, 5, 13, '2022-05-05 09:30:00',7), -- Finnick Kills Odin

(8, 3, 4, '2022-05-05 16:30:00',5), -- Cato Kills Rue

(9, 2, 14, '2022-05-05 16:56:00',5), -- Peeta Kills Felicity

(10, 5, 8, '2022-05-06 09:30:00',9), -- Finnick kills Thresh

(11, 9, 5, '2022-05-08 13:40:00',10), -- Beetee Killed Finnick

(12, 3, 9, '2022-05-09 14:15:00',6), -- Cato killed Beetee

(13, 1, 3, '2022-05-10 12:10:00',8); -- katniss killed Cato

INSERT INTO Weapons (Weapon\_ID, Weapon\_Type)

VALUES

(1,'Bow and Arrow'),

(2,'Long Knife'),

(3,'Sword'),

(4,'Trident'),

(5,'Spear'),

(6,'Mace'),

(7,'Sickle'),

(8,'Small Knife'),

(9,'Machete'),

(10,'Javelin'),

(11,'Katana'),

(12,'Blow Darts'),

(13,'Hatchet'),

(14,'Axe');

INSERT INTO Mentors (Mentor\_ID, FirstName, LastName, District\_ID)

VALUES

(1,'Haymitch','Abernathy', 1),

(2,'Nirali','Shah', 2),

(3,'Michaela','Wegg', 3),

(4,'Ceri','PattersonNairn', 4),

(5,'Carol','Bryce', 5),

(6,'Rachael','Rowbottom', 6),

(7,'Vanny','Coding', 7),

(8,'Zoe','Data', 8),

(9,'Eric','Nairn', 9),

(10,'Connor','Groden', 10),

(11,'Milo','Crisp', 11),

(12,'Aayush','Woosh', 12);

INSERT INTO Tribute\_Weapons ( Tribute\_ID, Weapon\_ID)

VALUES

(1,1),

(2,2),

(3,3),

(5,4),

(15,5),

(14,6),

(11,7),

(4,1),

(8,9),

(3,10),

(12,11),

(14,12),

(13,13),

(6,14),

(9,2),

(10,3),

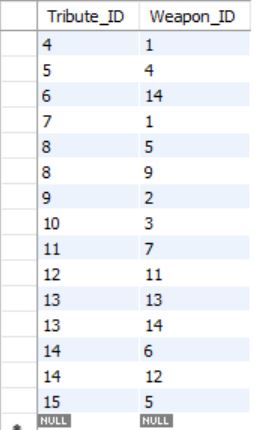
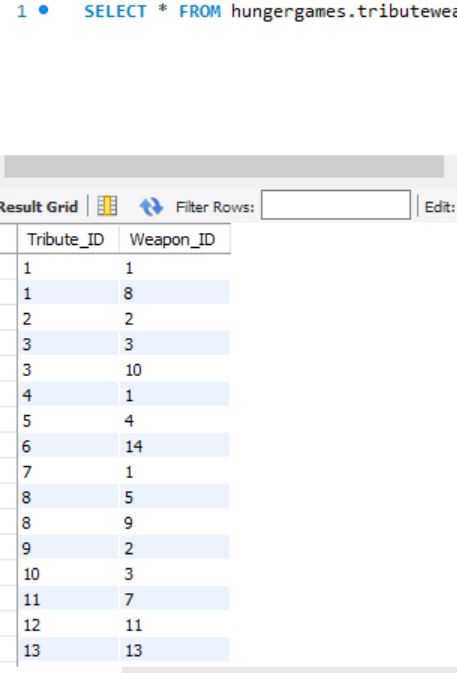
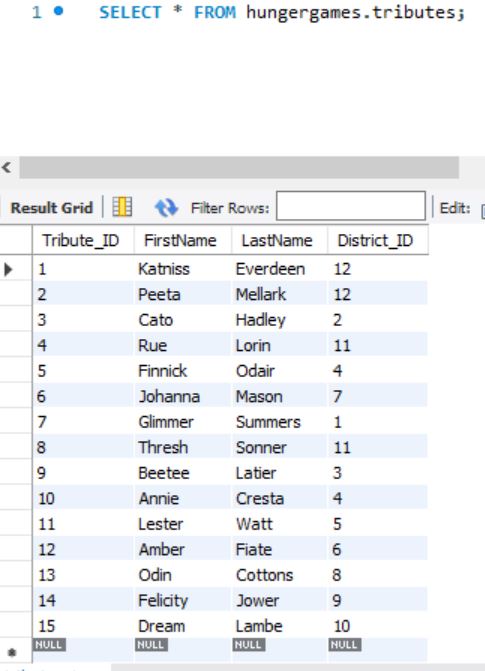
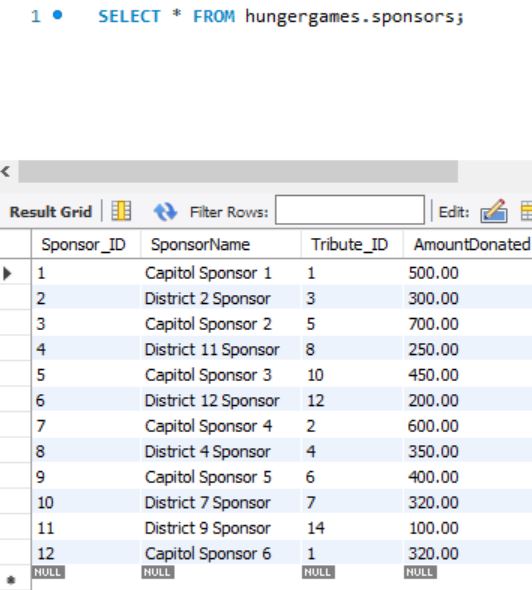
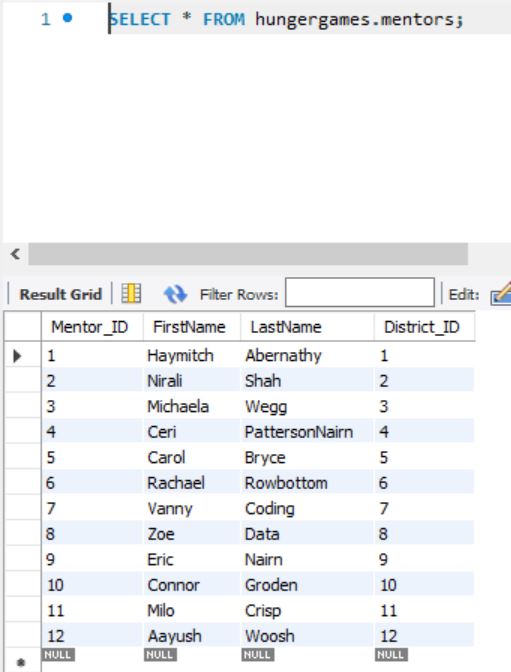
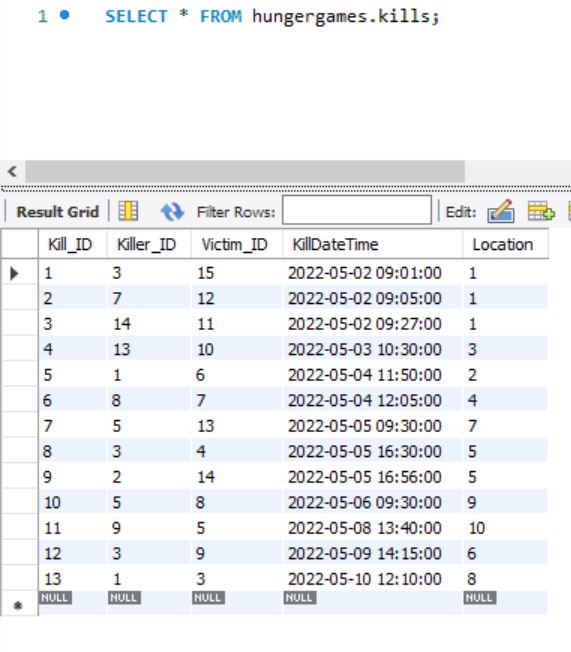
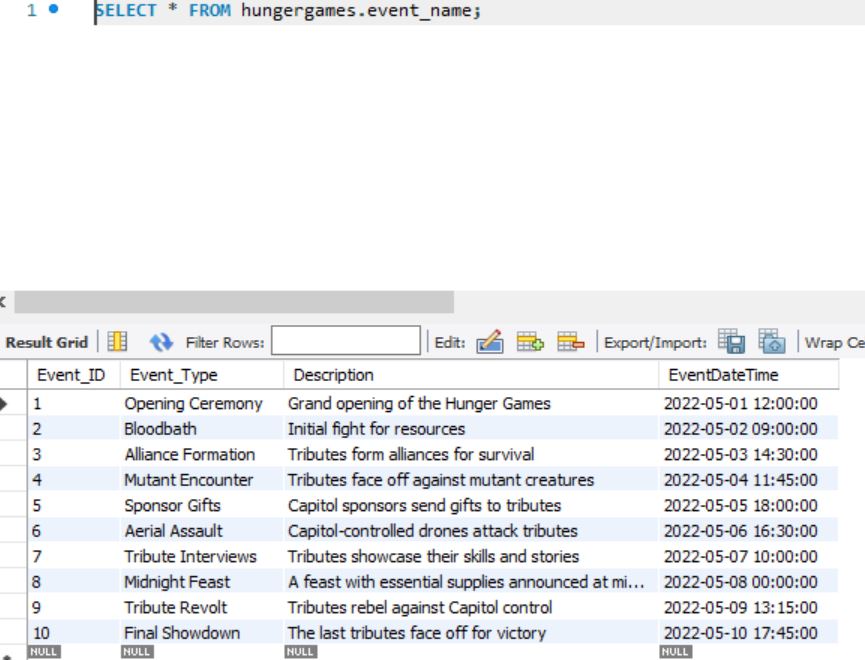
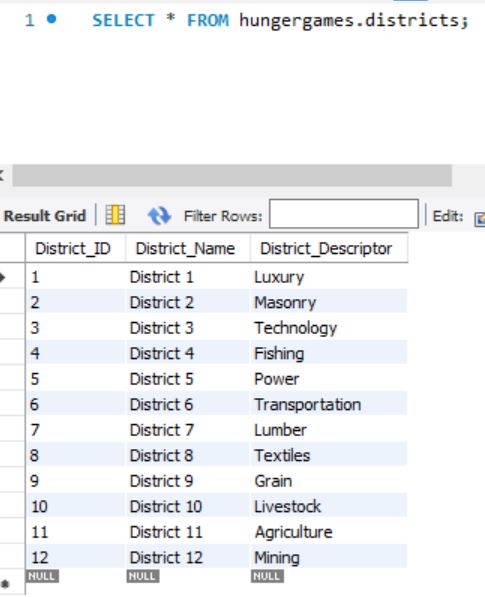
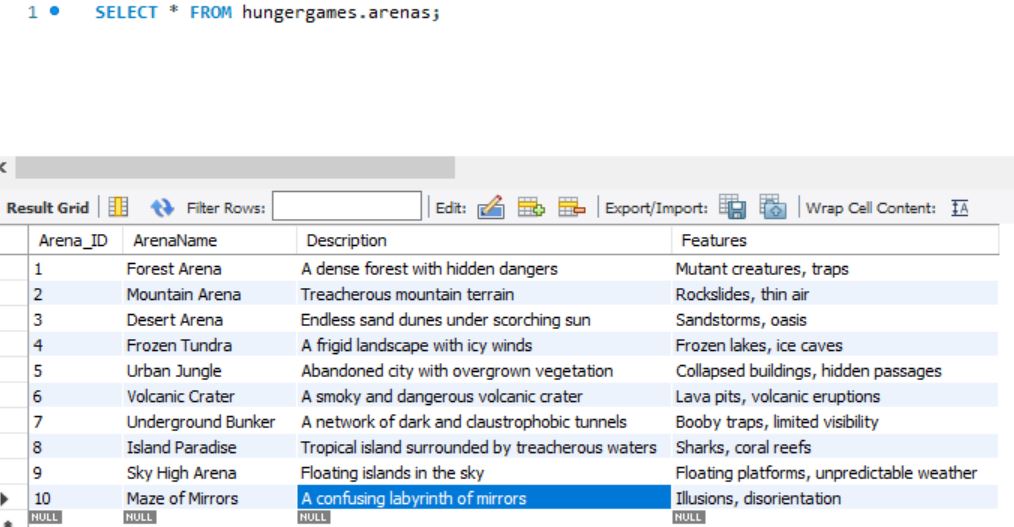
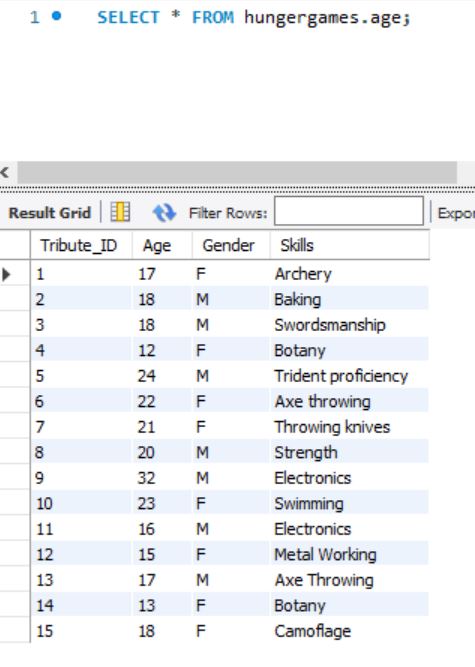
(8,5),

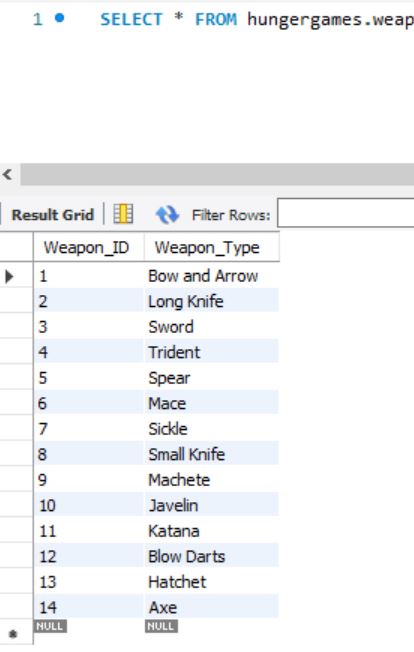
(1,8),

(7,1),

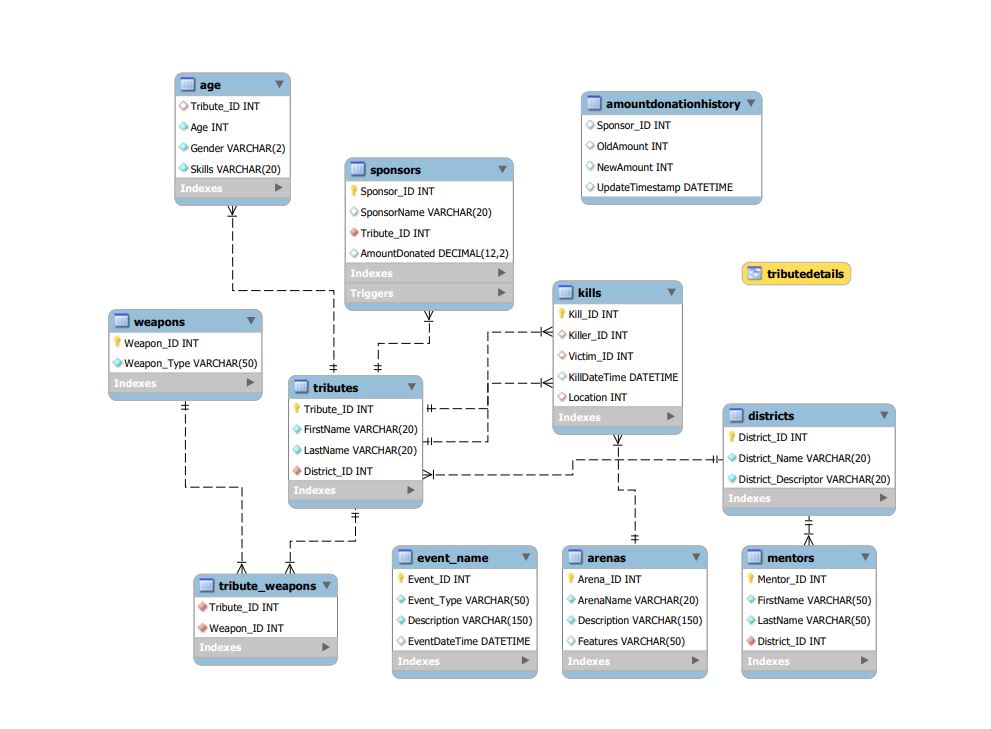
(13,14);

**Tables:**





**EER Diagram:**

****

**Create a view using multiple tables**

**Using five tables (‘Tributes’, ‘Districts’, ‘Age’, ‘Weapons’ and ‘Sponsors’):**

USE HungerGames;

CREATE VIEW TributeDetails AS

SELECT

tributes.Tribute\_ID,

tributes.FirstName,

tributes.LastName,

districts.District\_Name,

age.Age,

age.Gender,

age.Skills,

weapons.Weapon\_Type,

sponsors.SponsorName

FROM

tributes

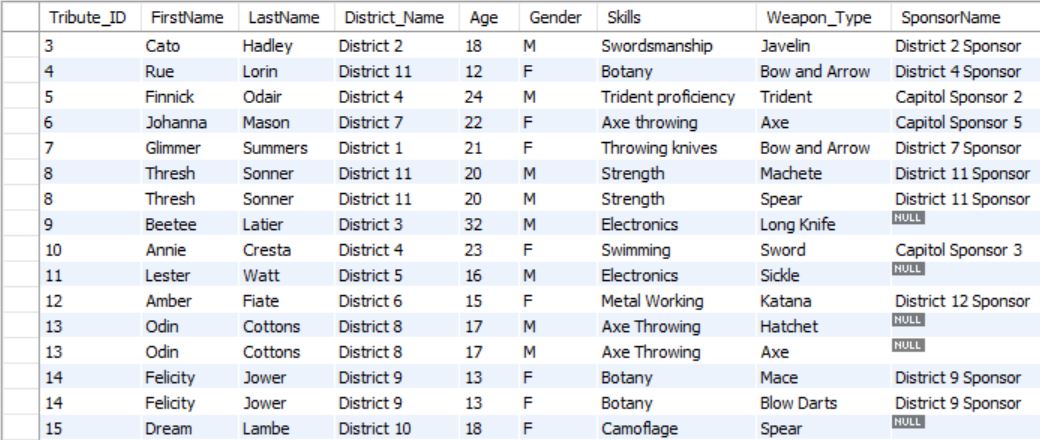
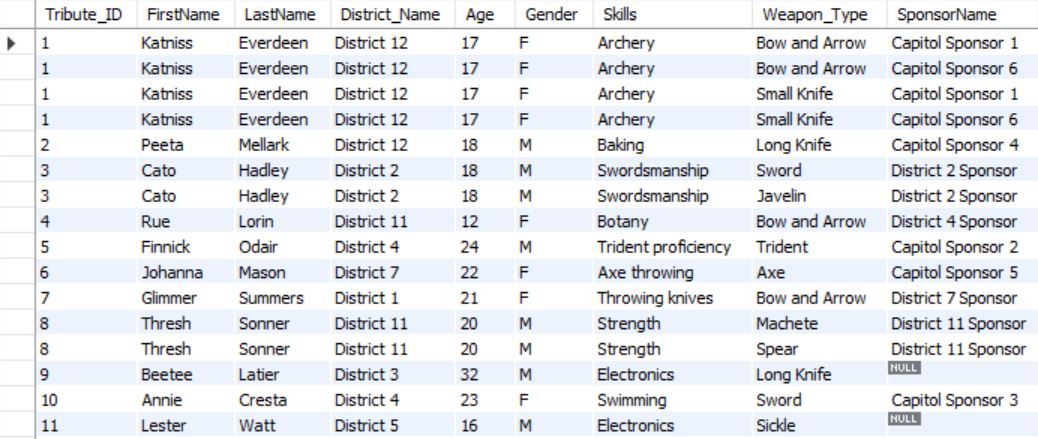
LEFT JOIN districts ON tributes.District\_ID = districts.District\_ID

LEFT JOIN age ON tributes.Tribute\_ID = age.Tribute\_ID

LEFT JOIN Tribute\_Weapons ON tributes.tribute\_ID = Tribute\_Weapons.Tribute\_ID

LEFT JOIN weapons ON Tribute\_Weapons.Weapon\_ID = weapons.Weapon\_ID

LEFT JOIN sponsors ON tributes.Tribute\_ID = sponsors.Tribute\_ID;



**Stored Procedure:**

DELIMITER //

CREATE PROCEDURE GetTributeStatus(IN p\_Tribute\_ID INT)

BEGIN

DECLARE v\_IsAlive BOOLEAN;

DECLARE v\_KillerName VARCHAR(50);

DECLARE v\_ArenaName VARCHAR(50);

DECLARE v\_EventName VARCHAR(50);

DECLARE v\_EventDateTime DATETIME;

-- Check if the tribute is alive

SELECT COUNT(\*) INTO v\_IsAlive

FROM Kills

WHERE Victim\_ID = p\_Tribute\_ID;

IF v\_IsAlive = 0 THEN

-- Tribute is alive

SELECT 'Alive' AS Status, FirstName FROM Tributes

WHERE Tribute\_ID = p\_Tribute\_ID;

ELSE

-- Tribute is dead, get details

-- below part is finding all the connections with the id number we have passed in

SELECT

tk.FirstName AS KillerName,

a.ArenaName,

en.Event\_Type AS EventName,

en.EventDateTime

INTO

v\_KillerName,

v\_ArenaName,

v\_EventName,

v\_EventDateTime

FROM Kills k

JOIN

Tributes tk ON k.Killer\_ID = tk.Tribute\_ID

JOIN

Arenas a ON k.Location = a.Arena\_ID

JOIN

Event\_Name en ON

en.EventDateTime = (

SELECT MAX(EventDateTime)

FROM Event\_Name

WHERE EventDateTime <= k.KillDateTime

)

-- the above subquery is checking the kill time with event time ranges. What it is doing is getting the last possible published time and then comparing it with kill time so if it is earlier than the last event time, then it is part of the earlier event and so on.

WHERE k.Victim\_ID = p\_Tribute\_ID

ORDER BY en.EventDateTime

LIMIT 1;

-- below part is returning for output

SELECT

'Dead' AS Status,

tk.FirstName AS Tribute,

v\_KillerName AS Killer,

v\_ArenaName AS Arena,

v\_EventName AS EventName

FROM Kills k

JOIN

Tributes tk ON k.Victim\_ID = tk.Tribute\_ID

WHERE

k.Victim\_ID = p\_Tribute\_ID;

END IF;

END //

--this procedure gives us in which arena was the tribute killed and who killed them, in case the tribute was alive at the end it will just give us the status alive

**Trigger:**

DELIMITER //

CREATE TRIGGER AfterUpdateAmountDonated

AFTER UPDATE ON Sponsors

FOR EACH ROW

BEGIN

IF NEW.AmountDonated <> OLD.AmountDonated THEN

INSERT INTO AmountDonationHistory (Sponsor\_ID, OldAmount, NewAmount, UpdateTimestamp)

VALUES (NEW.Sponsor\_ID, OLD.AmountDonated, NEW.AmountDonated, NOW());

END IF;

END//

DELIMITER ;

**Event:**

SET GLOBAL event\_scheduler = ON;

DELIMITER //

CREATE EVENT IF NOT EXISTS DailyDonationEvent

ON SCHEDULE

EVERY 1 DAY

STARTS TIMESTAMP('2022-05-01 00:00:00')

DO

BEGIN

UPDATE Sponsors

SET AmountDonated = AmountDonated + 100.00

WHERE SponsorName = 'Capitol Sponsor 1';

END //

DELIMITER ;

--This event is set to run every day, at midnight. It updates the `AmountDonated` for 'Capitol Sponsor 1', adding 100.00 to the existing amount.

**Stored Function:**

DELIMITER //

CREATE FUNCTION DonationStats(TributeID INT)

RETURNS INT

DETERMINISTIC

BEGIN

DECLARE totalDonation INT;

-- Initialize totalDonation to 0

SET totalDonation = 0;

-- Check if the Tribute\_ID exists in the Sponsors table

IF EXISTS (SELECT 1 FROM Sponsors WHERE Tribute\_ID = TributeID) THEN

-- Calculate the total donation amount for the specified Tribute\_ID

SELECT SUM(AmountDonated) INTO totalDonation

FROM Sponsors

WHERE Tribute\_ID = TributeID;

END IF;

RETURN totalDonation;

END;

DELIMITER ;

--This function gives us the total donated amount for tribute input. If the tribute doesnt have any donations( ie doesn’t exist in the sponsor table) the out is 0.

**How to call the stored procedure & stored function:**

CALL GetTributeStatus(2);

SELECT DonationStats(15) AS TotalDonationForTribute;

**Example Subquery:**

SELECT

t.Tribute\_ID,

t.FirstName,

t.District\_ID,

(

SELECT GROUP\_CONCAT(w.Weapon\_type SEPARATOR ', ')

FROM tribute\_weapons tw

JOIN weapons w ON tw.Weapon\_ID = w.Weapon\_ID

WHERE tw.Tribute\_ID = t.Tribute\_ID

) AS AssignedWeapons

FROM

tributes t;

--Uses 3 tables to get the weapons assigned to the tributes

**To run the Trigger:**

--Had to create a new table for this

USE hungergames;

CREATE TABLE AmountDonationHistory(

Sponsor\_ID INT,

OldAmount INT,

NewAmount INT,

UpdateTimestamp DATETIME

);

USE hungergames;

-- Perform an update on the Sponsors table to trigger the AfterUpdateAmountDonated event

UPDATE Sponsors

SET AmountDonated = AmountDonated + 50.00

WHERE Sponsor\_ID = 1;

**Groupby Query:**

SELECT

tw.Weapon\_ID,

w.Weapon\_type,

COUNT(t.Tribute\_ID) AS TributeCount

FROM

tribute\_weapons tw

JOIN

weapons w ON tw.Weapon\_ID = w.Weapon\_ID

JOIN

tributes t ON tw.Tribute\_ID = t.Tribute\_ID

GROUP BY

tw.Weapon\_ID, w.Weapon\_type;

--this uses group\_by to get number of tributes that have the same weapon

**Getting info out of the view:**

SELECT

District\_Name,

AVG(age) AS AverageAge,

MAX(age) AS MaxAge,

MIN(age) AS MinAge,

COUNT(DISTINCT Tribute\_ID) AS TotalTributes,

COUNT(DISTINCT SponsorName) AS TotalSponsors

FROM

tributedetails

GROUP BY

District\_Name;

--gives a lot of info using the view. - avg age, max age, min age, total tributes from the district, total sponsors for the district, everything is grouped by the district.