

University Of Asia Pacific

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (CSE)

**Course Title:** Database Management System **Course Code:** CSE-212

**Project**

On

**“July Uprising Management System”**

**Submitted by: Submitted to:**

**22201222** -Sajid Arefin Saad **Nadeem Ahmed**

**22201224** -Udoy Rebeiro Assistant Professor

**22201229 -** Sajjad Ullah Fahim Department of CSE

**22201230 -** Maisha Noor University of Asia Pacific

**22201247 -** Shahadat Hossen Riaz

**Sec : E1 (2.2)**

**Table of Content**

1. Topic ...........................................................................................3

2. Entities & Attributes....................................................................3

3. Relation between the Entities ....................................................6

4. ER Diagram ................................................................................10

5. Schema Diagram ........................................................................10

6. Attributes for Each table with keys ............................................11

7. Queries .......................................................................................12

8. View………………………………………………………………………………………..21

9.CEP Mapping ...............................................................................23

**July Uprising Management System**

**Topic –**

The July Uprising Management System is designed to efficiently organize and manage information related to large-scale protests and social movements. This database tracks and integrates key aspects of events, including details of individuals involved, organizations, media coverage, locations, medical facilities, injuries, arrests, protests, and social media posts. The system facilitates the documentation and analysis of events by maintaining relationships between entities like protests and media, injuries and events, and medical facilities and locations.

The ER diagram represents the database schema for the **July Uprising Management System**, which is designed to document and manage data related to large-scale student-led protests in Bangladesh. Below is a summarized explanation of its components and relationships.

**Entities and their attributes :**

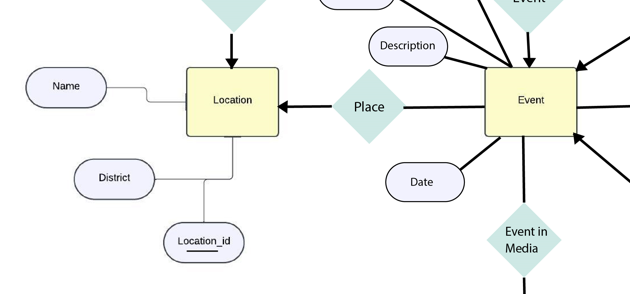
* People: Tracks individuals participating in the movement, including their roles and information.
* P\_Id (Primary Key)
* Name
* Age
* Role
* Event: Maintains records of significant events, protests, campaigns, and meetings, along with their descriptions and dates.
* Event\_Id (Primary Key)
* Name
* Description
* Date
* Injuries: Monitors injuries during protests, categorizing them by severity.
* Injury\_id (Primary Key)
* Severity
* Organizations: Captures details about different organizations and their type.
* Org\_Id (Primary Key)
* Name
* Type
* Arrest: Documents arrests made during protests, reasons for detention, and durations of custody
* Arrest\_id (Primary Key)
* Reason
* Duration
* Media: Stores details about photos, videos, audio recordings, and graphics documenting the protests and their impact.
* Media\_Id (Primary Key)
* Type
* Description
* Location: Stores information about important locations across various districts where events happened.
* Location\_id (Primary Key)
* Name
* District
* Medical facility: Records healthcare facilities providing aid to injured protesters, including their locations and capacities.
* M\_id (Primary Key)
* Name
* Capacity
* Social media Post: Tracks social media posts related to the movement, identifying platforms, authors, and dates of publication.
* Post\_id (Primary Key)
* Author
* Platform
* Date

**Relations between the entities :**

**Location and Event Relationship:**

Specifies the location where an event occurs.

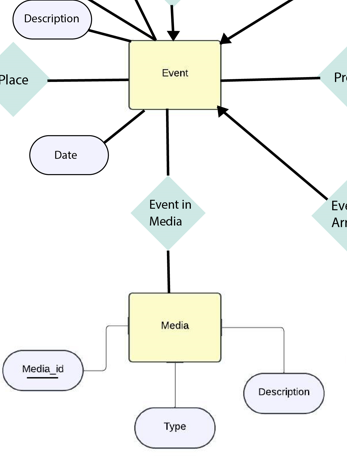
Type: One-to-Many (A location can have multiple events, but an event happens in one location).



**Event and Media Relationship:**

Highlights the media coverage of an event.

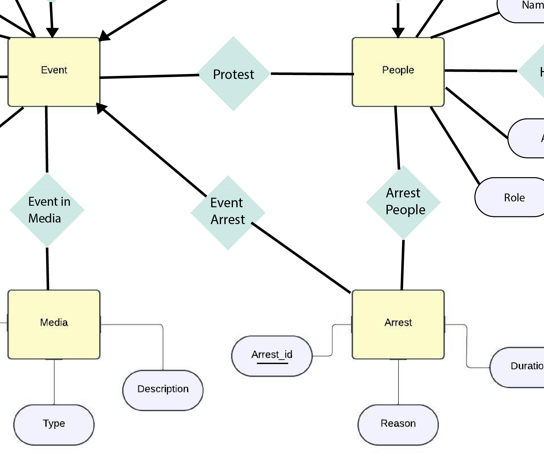
Type: Many-to-Many (An event can appear in multiple media outlets, and media outlets can cover multiple events).



**Event and Arrest Relationship:**

Indicates arrests made in connection with an event.

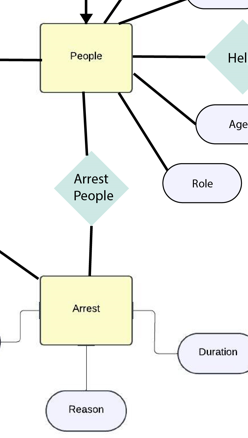
Type: One-to-Many (An event can lead to multiple arrests).



**Arrest and People Relationship:**

Links arrested people with specific arrests.

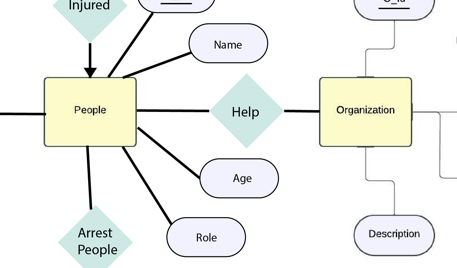
Type: One-to-Many (A person can be arrested multiple times).



**People and Organization Relationship**:

Describes support provided by organizations to people.

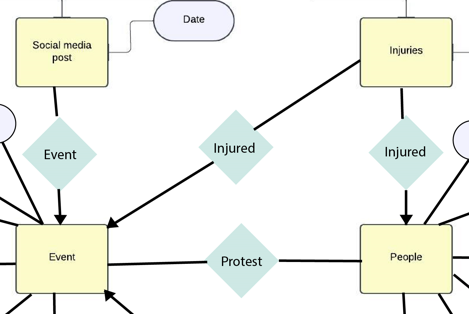
Type: One-to-Many (One people can connected to multiple Organizations).



**Event and Injuries Relationship**:

Describes injuries caused during an event.

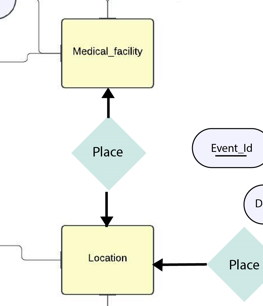
Type: One-to-Many (An event can result in multiple injuries).



**Location and Medical Facility Relationship:**

Specifies the location of medical facilities.

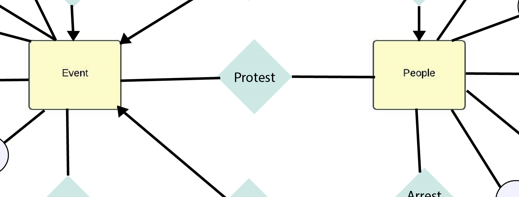
Type: One-to-One (Each medical facility is located in a specific location).



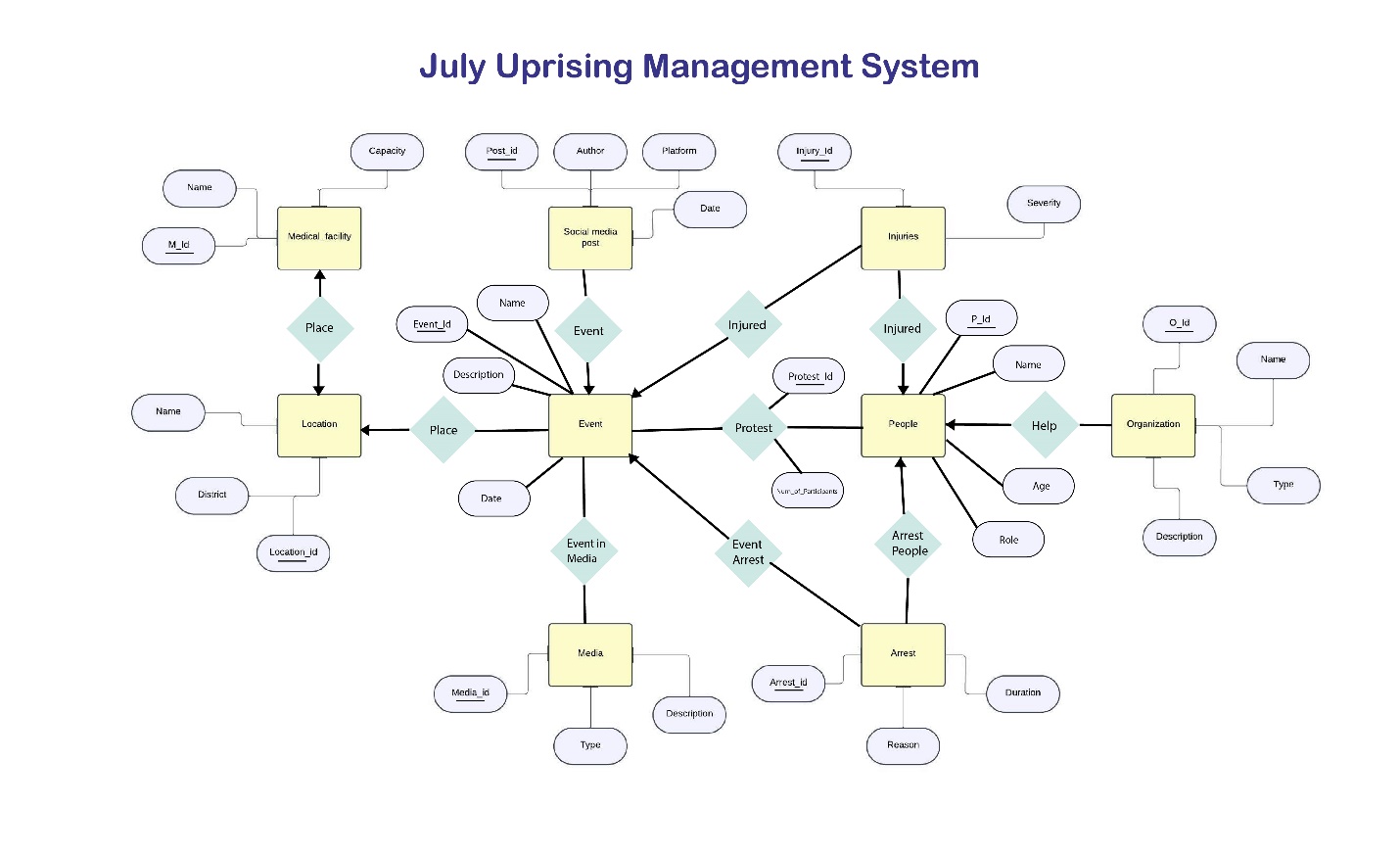
**Event and People Relationship:**

Represents the involvement of individuals in events.

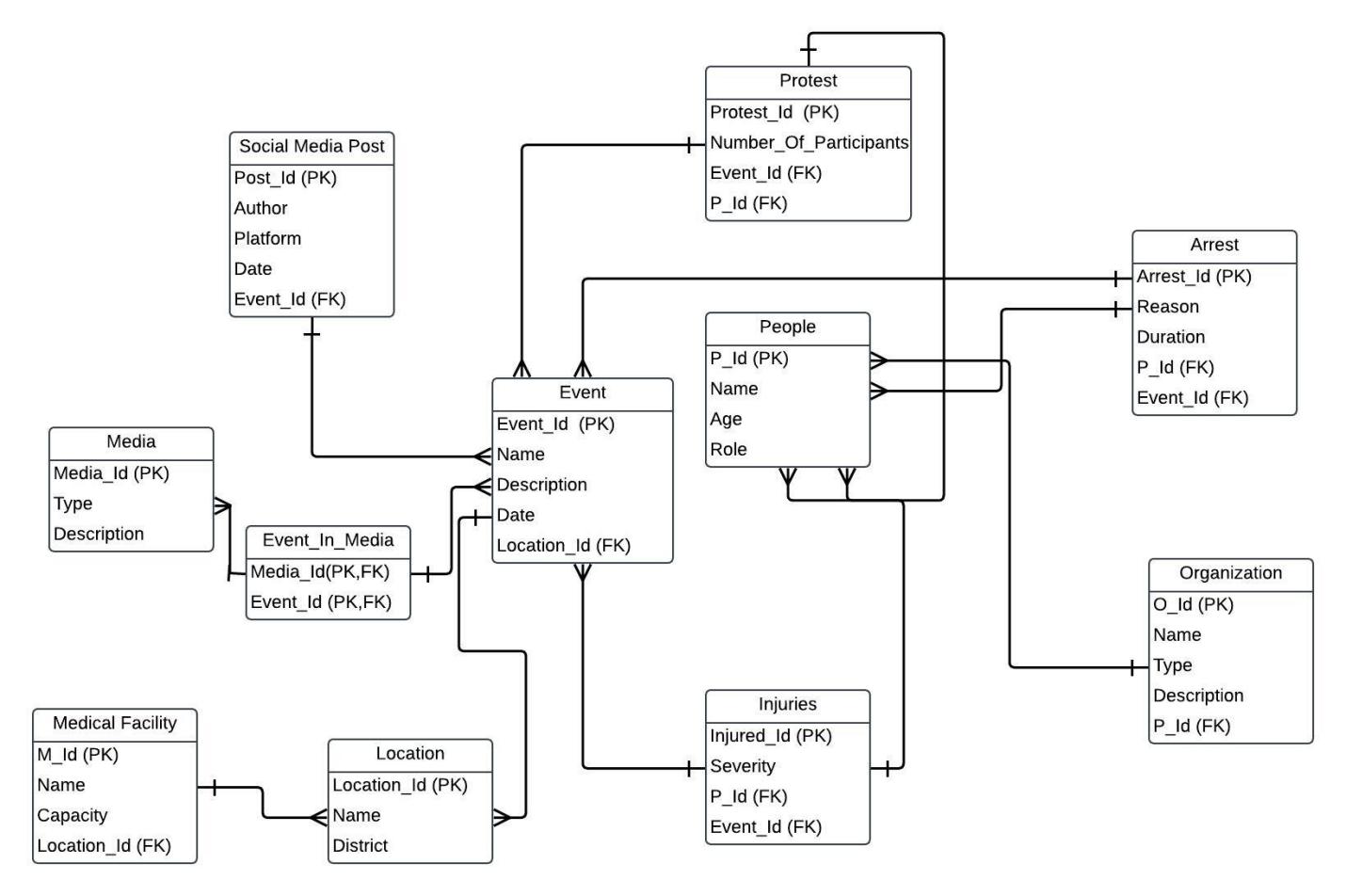
Type: Many-to-Many (An event can involve multiple people in different roles, and a person can be part of multiple events)



**Er Diagram:**

* 

**Schema Diagram:**



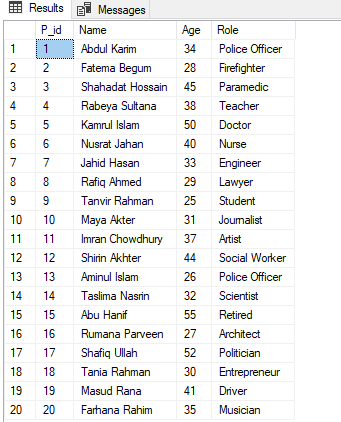
**Attributes for Each table with keys :**

* People:
* P\_Id (Primary Key)
* Name
* Age
* Role
* Event:
* Event\_Id (Primary Key)
* Name
* Description
* Date
* Location\_id (Foreign Key)
* Injuries:
* Injury\_id (Primary Key)
* P\_id (Foreign Key)
* Event\_Id (Foreign Key)
* Severity
* Organizations:
* Org\_Id (Primary Key)
* Name
* Type
* Description
* P\_Id(Foreign Key)
* Arrest:
* Arrest\_id (Primary Key)
* P\_id (Foreign Key)
* Event\_id (Foreign Key)
* Reason
* Duration
* Media:
* Media\_Id (Primary Key)
* Type
* Description
* Location:
* Location\_id (Primary Key)
* Name
* District
* Medical facility:
* M\_id (Primary Key)
* Name
* Location\_id (Foreign Key)
* Capacity
* Social media Post:
* Post\_id (Primary Key)
* Author
* Event\_id (Foreign Key)
* Platform
* Date
* Protest:
* Protest\_id (Primary Key)
* P\_id (Foreign Key)
* Event\_Id(Foreign Key)
* Number of participants
* Event\_In\_Media:
* Protest\_id (Primary Key, Foreign Key)
* P\_id (Primary Key, Foreign Key)

**Queries:**

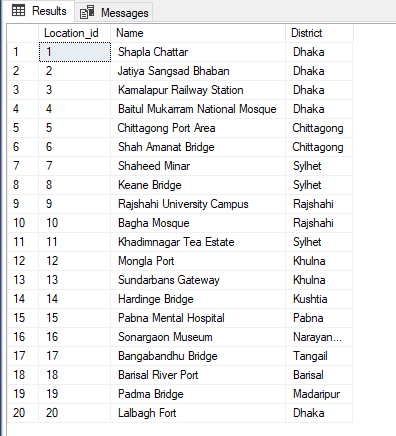
1. Print all the data of People table:

select \* from people



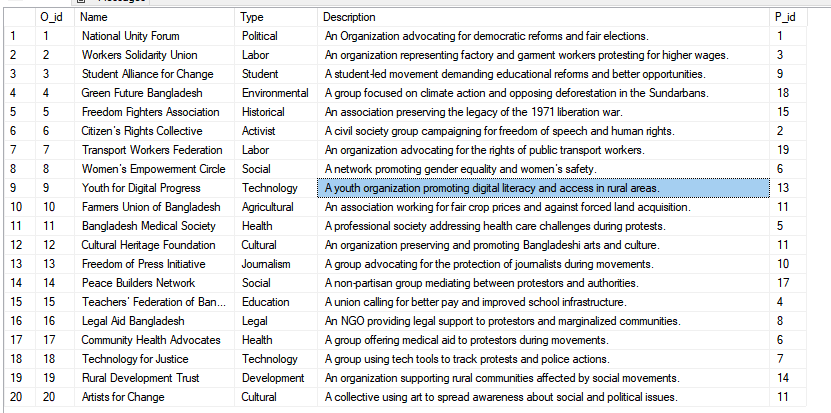
2.Print all the data of Location table:

select \* from Location



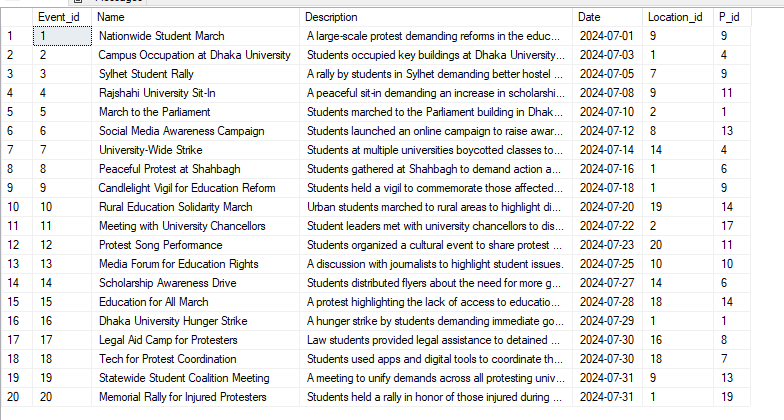
3.Print all the data of Organization table:

select \* from Organization



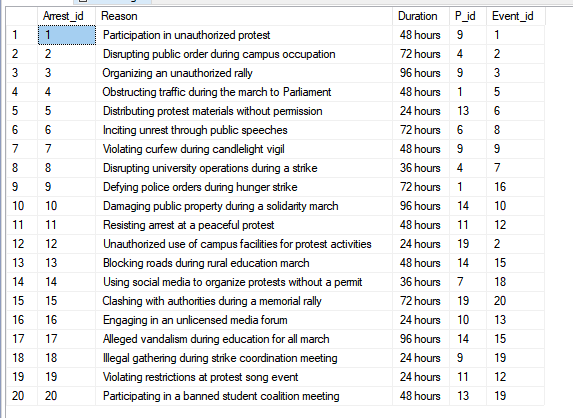
4.Print all the data of Event table:

select \* from Event



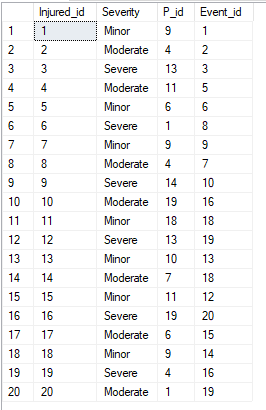
5.Print all the data Arrest table:

select \* from Arrest



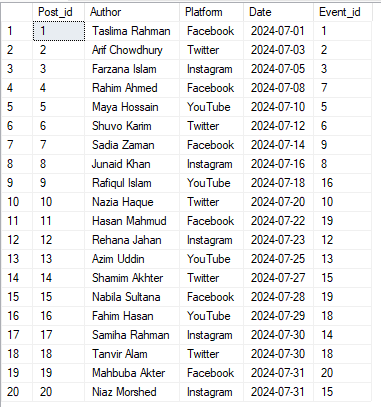
6.Print all the data of Injuries table:

select \* from Injuries



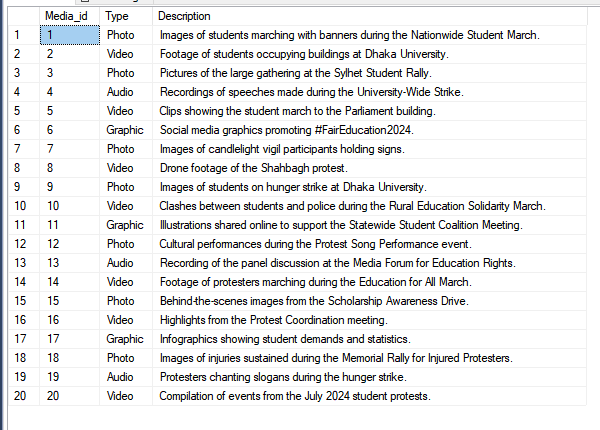
7.Print all the data of Social\_Media\_Post table:

select \* from Social\_media\_Post



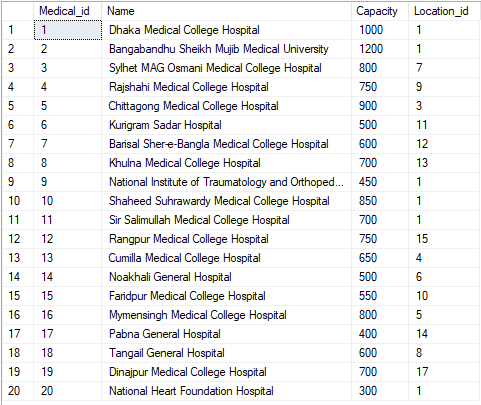
8.Print all the data of Media table:

select \* from Media



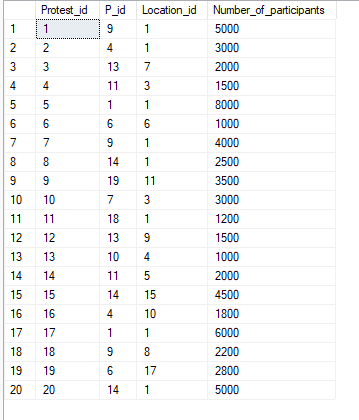
9.Print all the data of Medical\_Facility table:

select \* from Medical\_facility



10.Print all the data of Protest table:

select \* from Protest

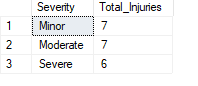


11. Count the total number of injuries by severity.

SELECT Severity, COUNT(\*) AS Total\_Injuries

FROM Injuries

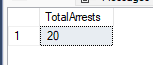
GROUP BY Severity;



12. Find the total number of arrests made during the protests

SELECT COUNT(\*) AS TotalArrests

FROM Arrest;



13. Combine lists of people who were either arrested or injured during the protests

SELECT Name

FROM People

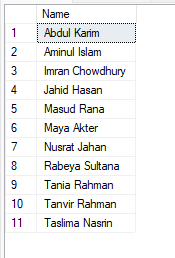
WHERE P\_Id IN (SELECT P\_Id FROM Injuries)

UNION

SELECT Name

FROM People

WHERE P\_Id IN (SELECT P\_Id FROM Arrest);



14. Find people who participated in protests and were also injured during events

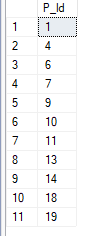
SELECT P\_Id

FROM Protest

INTERSECT

SELECT P\_Id

FROM Injuries;



15. List all names of locations where events occurred or where medical facilities are available

SELECT Name

FROM Location

WHERE Location\_Id IN (SELECT Location\_Id FROM Event)

UNION

SELECT Name

FROM Location

WHERE Location\_Id IN (SELECT Location\_Id FROM Medical\_Facility);

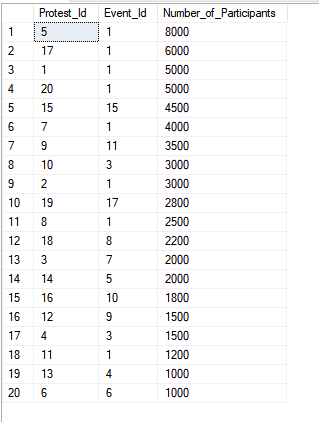


16. List protests ordered by the number of participants in descending order.

SELECT Protest\_Id, Event\_Id, Number\_of\_Participants

FROM Protest

ORDER BY Number\_of\_Participants DESC;

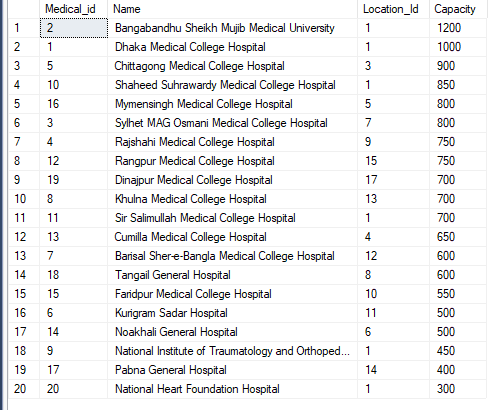


17. List medical facilities by capacity descending , then by name alphabetically

SELECT Medical\_id, Name, Location\_Id, Capacity

FROM Medical\_Facility

ORDER BY Capacity DESC, Name ASC;

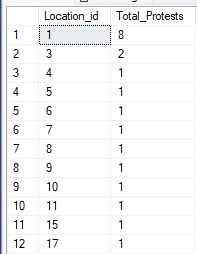


18.Count the number of protests at each location

SELECT Location\_id, COUNT(Protest\_Id) AS Total\_Protests

FROM Protest

GROUP BY Location\_id;



19. Find people involved in more than 2 arrests

SELECT P\_Id, COUNT(Arrest\_Id) AS Total\_Arrests

FROM Arrest

GROUP BY P\_Id

HAVING COUNT(Arrest\_Id) > 2;



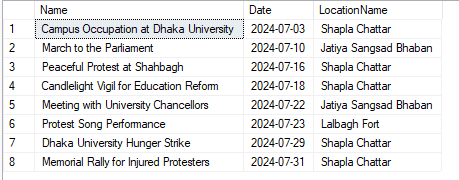
20. List all events that occurred in Dhaka.

SELECT E.Name, E.Date, L.Name AS LocationName

FROM Event E, Location L

WHERE E.Location\_id = L.Location\_id

AND L.District = 'Dhaka';



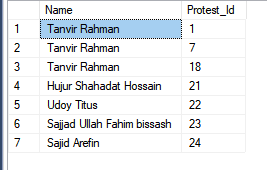
1. Get a list of all protests led by students

SELECT Pe.Name,P.Protest\_Id

FROM People Pe, Protest P

WHERE Pe.P\_id = P.P\_id

AND Pe.Role = 'Student';



**View :**

**Create a View for Events Involving Students**

CREATE VIEW StudentEvents AS

SELECT E.Name AS EventName, E.Date AS EventDate

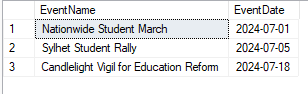
FROM Event E, People P

where E.P\_id = P.P\_id

and P.Role = 'Student'

TO see the view:

SELECT \* FROM StudentEvents;



**Create a view to combine events held in Dhaka and Chittagong**

CREATE VIEW DhakaAndChittagongEvents AS

SELECT Name, Date

FROM Event

WHERE Location\_id IN (SELECT Location\_id FROM Location WHERE District = 'Dhaka')

UNION

SELECT Name, Date

FROM Event

WHERE Location\_id IN (SELECT Location\_id FROM Location WHERE District = 'Chittagong');

TO see the view:

SELECT \* FROM DhakaAndChittagongEvents;



**Complex Engineering Problem Mapping**

Our project is a solution to a complex engineering problem because it requires specialized knowledge in database systems and advanced query optimization. Resolving the problem involves designing an efficient database schema, writing optimized SQL queries, and applying abstract thinking to accommodate relationships between entities such as events, locations, and protests. There is no single obvious solution, as the structure of the database and the requirements of the query significantly influence the approach. It demands in-depth understanding of database normalization and execution plans to handle potentially large datasets effectively and ensure accurate results.

**How Ks are addressed through the project and mapping among Ks, Cos and Pos**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ks | Attribute | How Ks are addressed through the project | Related Cos | Related Pos |
| K3 | Engineering Fundamentals | Applied core engineering principles (database design, SQL, and relational database modeling) to develop this system. | CO1,CO2 | 5(e),2(b) |
| K5 | |  |  | | --- | --- | |  | Engineering Design | | |  | | --- | | Designed ER diagrams and schema for tracking of events, social media activity, and associated entities like people and organizations. |      |  | | --- | |  | | CO3,CO4 | 1(a),2(b) |
| K6 | |  | | --- | | Engineering Practice |      |  | | --- | |  | | We have used SQL in Microsoft SQL Server and Illustrator for making ER diagram. | CO1,CO2 | 5(e),2(b) |

**How Ps are addressed through the project and mapping among Ps, Cos and Pos**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ps | Attribute | How Ps are addressed through the project | Related Cos | Related Pos |
| P1 | Depth of knowledge required | The project requires study of SQL in Microsoft SQL Server(K6), normalization, ER diagram, Schema diagram(K5), join, subquery etc. | CO2,CO4 | 5(e),2(b),3(c) |
| P2 | Range of Conflicting Requirement | Designed systems to address real-world challenges (e.g., tracking people, media posts, and protests) with queries. | CO5 | 3(c) |
| P3 | Depth of Data Analysis | Designed tables to store, process and analyze data. Created queries to analyze and predict patterns in tables. | CO5 | 3(c) |
| P7 | Interdependence | Sub problems have physical data independency | CO8 | 9(i) |

**How As are addressed through the project and mapping among As, Cos and Pos**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| As | Attribute | How As are addressed through the project | Related Cos | Related Pos |
| A1 | Range of Resources | Utilized diverse resources online platforms, social media inputs and public records, for mapping the project effectively. Money | C05, CO9 | 3(c), 10(j) |
| A2 | Level of Interaction | |  | | --- | |  |  |  | | --- | | Enabled high interaction through a database system that integrates real-time social media posts and user inputs to validate protest data | | CO8,CO9 | 9(i),10(j) |
| A3 | Innovation | |  | | --- | |  |  |  | | --- | | Designed and implemented a system that applies query mechanisms for dynamic data handling. | | CO5 | 3(c) |
| A5 | Familiarity | The project explores event mapping with familiar concepts like database schema design and public data integration, ensuring accessibility for end-users. | CO8 | 9(i) |

|  |
| --- |
|  |