The BRITISH UNIVERSITY IN EGYPT

Computer Engineering Department

Database Systems

22COMP06I

Project Report

Submitted to

Dr. Abir Ali & Eng. Amr

ELEC. Dep., BUE

Students:

Mohamed Ahmed Abdel-Galeil	***ID:** 214618*
Mostafa Mohamed Ahmed	***ID:** 212577*
Mohamed Abdel-Moneim	***ID:** 210210*

COMP. Dep. DY2

Specification and Requirements

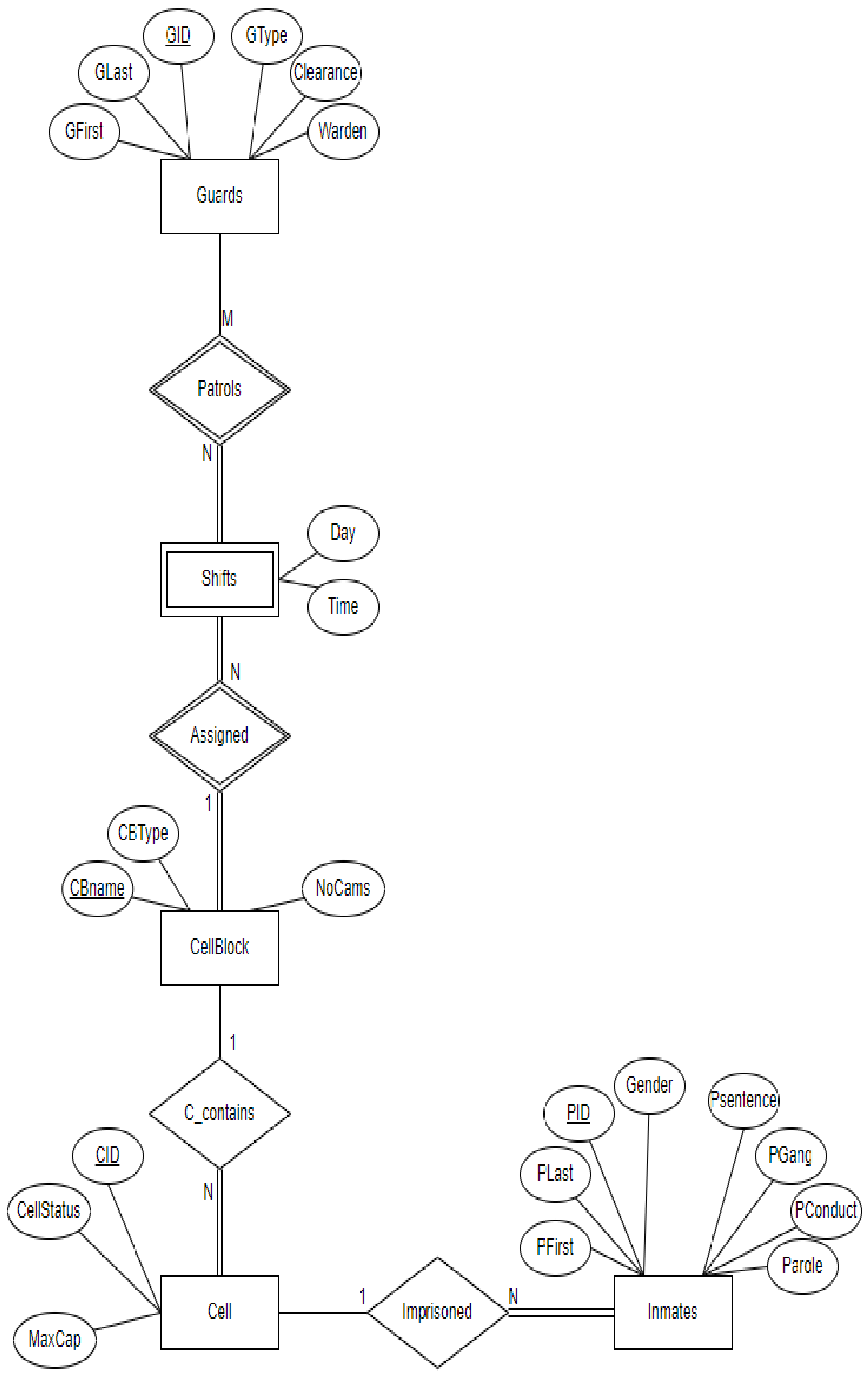
- The Prison Database consists of Cell Blocks. Each CellBlock has a unique name (CBname), a Cell Block Type (CBType), and a number of cameras (NoCams). The cellblocks are divided into 2 types, normal and solitary. ***Each Cell assigned to a solitary Cell Block has a maximum cap of 1***

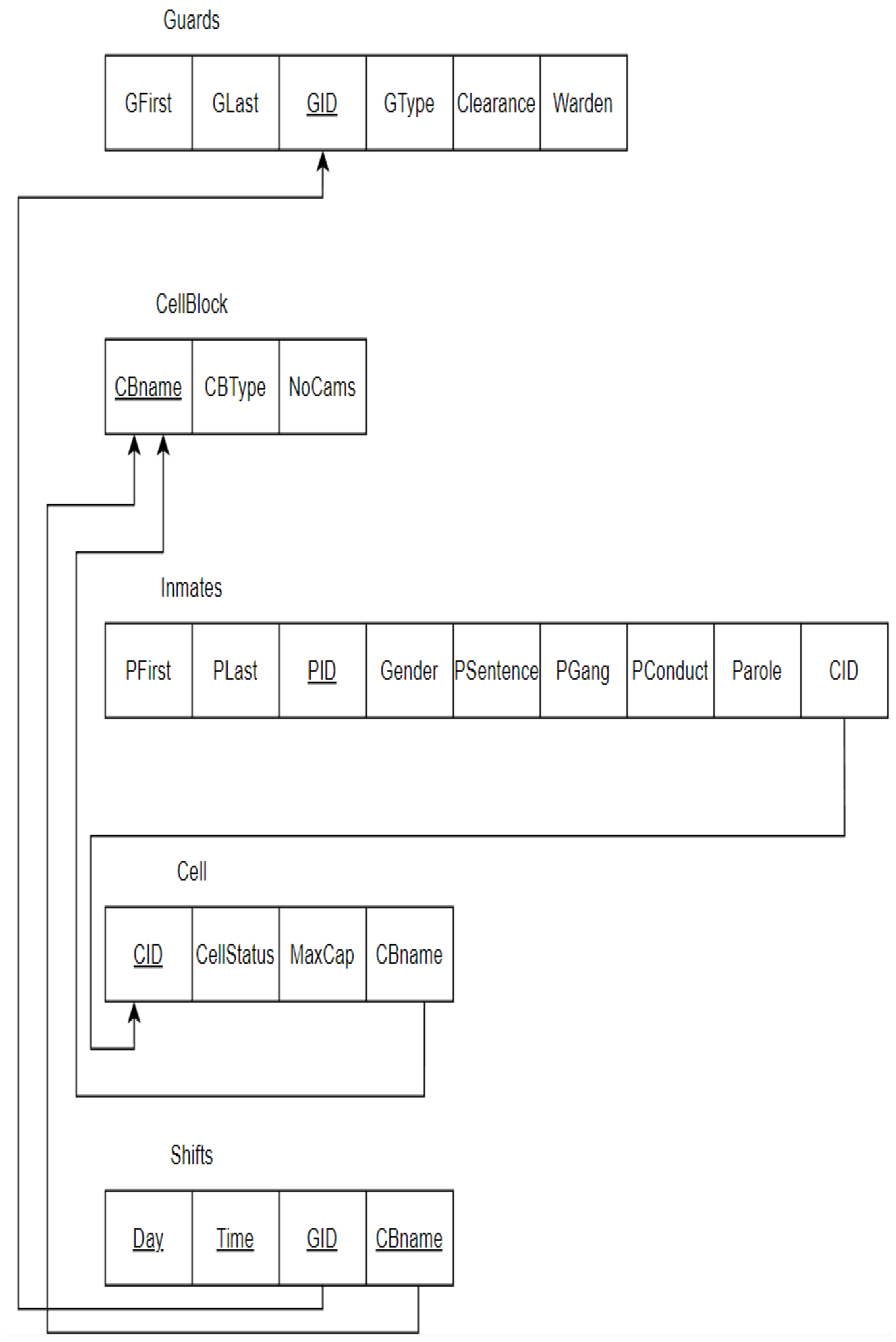
- Each Cell Block has a number of Shifts assigned to it, with each Shift consisting of a Day (Day) and a Time (Time). The Day determines which day of the week is assigned, and the Time determines if the shift occurs during the daytime or nighttime.

- The Guards table has a Unique ID (GID), a First name (GFirst), a Last name (GLast), a clearance level (Clearance), a Guard Type (GType), and a Warden Boolean (Warden). The guards could be a Warden, a Security Officer, a K-9 unit, a Riot unit, a Watchtower, or a security camera Operator. Since each guard type is assigned a specific duty, they do have a varying clearance level. The levels range from 0 to 4, with 0 being the highest. The warden has a clearance level of 0, the K-9 and the riot units have a clearance level of 1, the watch Tower guards have a clearance level of 2, the Security Camera Operators have a clearance level of 3, and the Security officers have a clearance level of 4***.*** The Guards patrol assigned shifts according to their Type, so some of them do not have a patrol designation. ***The Prison can only have 1 Warden.***

- Each Inmate is assigned a cell, which is assigned to a Cell Block. The database will store the following information regarding each Inmate: First name (PFirst), last name (PLast), prisoner ID (PID), Gender (Gender), Parole Eligibility (Parole), Prisoner Sentence (Psentence), Conduct (PConduct), Gang affiliation (PGang), and a designated cell block. The conduct of an inmate affects other factors including his eligibility for parole.

- Each Cell has a Cell ID (CID), Cell Status (CellStatus), maximum Cap (MaxCap), and Cell Block name (CBName).


GID
GType
GLast
Clearance
GFirst
Warden
Guards
M
Patrols
N
Day
Shifts
Time
N
Assigned
1
CBType
CBname
NoCams
CellBlock
1
C_contains
N
Gender
PID
Psentence
PGang
PLast
PConduct
PFirst
Parole
CID
CellStatus
Cell
1
Imprisoned
N
Inmates
MaxCap


Guards
GFirst
GLast
GID
GType
Clearance
Warden
CellBlock
CBname
CBType
NoCams
Inmates
PFirst
PLast
PID
Gender
PSentence
PGang
PConduct
Parole
CID
Cell
CID
CellStatus
MaxCap
CBname
Shifts
Day
Time
GID
CBname

Inmates								
PFirst	PLast	PID	Gender	PSentence	PGang	PConduct	Parole	CID
Jonathan	Khaled	10	M	25 Years	Cartel	Good	No	10000
Steven	Mohamed	20	M	10 Years	Null	Good	Yes	20000
Jordan	Peterson	30	M	Death Sentence	Null	Good	No	30000
Maria	Marely	40	F	50 Years	Mafia	Bad	No	50000
Stephanie	Hassan	50	F	1 Year	Null	Bad	No	50000

Guards					
GFirst	GLast	GID	GType	Clearance	Warden
Hossam	Ashraf	100	Security Officer	4	No
Tana	Alaa	101	Security Officer	4	No
Youssef	Yasser	102	Security Officer	4	No
Pablo	Mido	200	CCTV Operator	3	No
Khaled	Hany	300	Warden	0	Yes
Sara	Adel	400	Riot Unit	1	No
Carla	Jackson	500	K-9 Unit	1	No
Shady	Bibo	600	WatchTower	2	No

CellBlock		
CBname	CBType	NoCams
CB_A	Normal	5
CB_B	Normal	5
CB_C	Solitary	15
CB_D	Solitary	15

Cell			
CID	CBname	CellStatus	MaxCap
10000	CB_A	Full	4
20000	CB_B	Occupied	3
30000	CB_C	Maintenance	1
40000	CB_D	Empty	1

Shifts			
Day	Time	GID	CBname
Wednesday	Day	100	CB_A
Tuesday	Night	101	CB_B
Sunday	Day	101	CB_C
Friday	Night	102	CB_A

2 Simple queries based on one or more input	• Select all inmates with years more than the amount inputted by the user • Select all Guards with clearance level higher than the amount the user inputted
2 Queries that required data across more than two tables	• Select all the inmates specified by the Cell Block Type input • Select all Cell Block Types with the inputted value of Guards ID
2 Calculating queries	• Count number of shifts on each cellblock • Count the number of cells in the cellblock picked by the user.
1 query with pattern search in an attribute	• Find the CBType of all Cell Blocks that contain the keyword written by the user
2 Aggregating queries	• Get the average number of Guards in each Cellblock • For each Cell, Count the number of inmates who are assigned to it
1 union or intersection query	• Get IDs of cells placed in a certain cellblock or has MaxCap of a certain value, both of which are decided by the user
1 query to show that data that doesn't exist	• Get the ID of cells with no inmates.
1 Trigger	• If there is no guard with the Type Warden, assign a random security officer as a Warden Type

2 Simple queries based on one or more input	• SELECT * FROM Inmates WHERE PSentence > 25; • SELECT * FROM Guards WHERE Clearance > 3;
2 Queries that required data across more than two tables	• SELECT Inmates.* FROM Inmates INNER JOIN Cell ON Inmates.CID = Cell.CID INNER JOIN CellBlock ON Cell.CBname = CellBlock.CBname WHERE CellBlock.CBType = 'Normal_M'; • SELECT CellBlock.CBType FROM CellBlock INNER JOIN Shifts ON CellBlock.CBname = Shifts.CBname INNER JOIN Guards ON Shifts.GID = Guards.GID WHERE Guards.GID = 100;
2 Calculating queries	• SELECT CBname, COUNT(*) AS num_shifts FROM Shifts GROUP BY CBname; • SELECT COUNT(*) AS NoCells FROM Cell WHERE CBname = 'CB_A';
1 query with pattern search in an attribute	• SELECT * FROM CellBlock WHERE CBType LIKE '%Normal%';
2 Aggregating queries	• SELECT Shifts.CBname, Count(*) AS NoGuards From Guards JOIN Shifts ON Guards.GID = Shifts.GID GROUP BY(Shifts.CBname); • SELECT Cell.CBname, Cell.CID, COUNT(*) AS NoInmates FROM Inmates JOIN Cell ON Inmates.CID = Cell.CID GROUP BY Cell.CBname, Cell.CID;
1 union or intersection query	• SELECT CID FROM Cell WHERE CBname = 'CB_A' OR MaxCap = 3;
1 query to show that data that doesn't exist	• SELECT Cell.CID FROM Cell LEFT JOIN Inmates ON Cell.CID = Inmates.CID

	WHERE Inmates.CID IS NULL;

1 Trigger	• UPDATE Guards SET GType = 'Warden'
	WHERE GType = 'Security officer' AND Clearance = 0
	AND NOT EXISTS (SELECT 1 FROM Guards WHERE type = 'Warden');

2 Simple queries based on one or more input	<ul><li>This is a SQL query that selects all columns from the table Inmates and joins it with the table Cell on the condition that the CID column of both tables are equal. Then it filters the result by selecting only those rows where the value of the CBname column in the Cell table is equal to user defined value. The query is used to retrieve information about inmates who are in CellBlock ‘CB_A’.</li><li>This SQL query retrieves data from a table called "Guards" based on a condition specified in the "WHERE" clause. The condition is that the value of the "Clearance" column should be greater than a parameter denoted by a question mark in the query. The question mark acts as a placeholder for user input, which will replace it at runtime. The "SELECT" statement retrieves all columns from the "Guards" table for the rows that satisfy the above condition.</li></ul>
2 Queries that required data across more than two tables	<ul><li>This is a SQL query that retrieves information about inmates who are housed in cells that belong to a user specified type cell block. The query joins data from the "Inmates," "Cell," and "CellBlock" tables based on the "CID" and "CBname" columns. The query selects all columns from the "Inmates" table for the inmates who satisfy the condition that their cell block is of the user specified type.</li><li>This is a SQL query that retrieves information about the type of cell block where a guard with a specific ID (GID) is assigned to work. The query joins data from the "CellBlock," "Shifts," and "Guards" tables based on the "CBname" and "GID" columns. The query filters the results based on the condition that the "GID" column should be equal to the user specified input and retrieves the "CBType" column from the "CellBlock" table.</li></ul>
2 Calculating queries	<ul><li>This query is a SQL SELECT statement that selects the column “CBname” and counts the number of rows in the “Shifts” table for each unique value of “CBname”. The result of this query will be a table with two columns, “CBname” and “num_shifts”, where “CBname” is a unique value from the “Shifts” table and “num_shifts” is the number of rows in the “Shifts” table that have that value for “CBname”.</li><li>This SQL query selects the number of cells in the "Cell" table that are associated with the user inputted shift block. The "COUNT(*)" function counts the number of rows that match the condition specified in the "WHERE" clause. In this case, it counts the number of cells where the "CBname" column is equal to the user input. The "AS NoCells" statement renames the result column as "NoCells" for clarity.</li></ul>

1 query with pattern search in an attribute	• This is a SQL query that selects all the rows from a table called "CellBlock" where the value of the "CBType" column contains the user specified keyword anywhere within it.
2 Aggregating queries	• This SQL query selects the number of guards assigned to each shift block in the "Shifts" table, by joining the "Guards" table using the common GID (Guard ID) column, and grouping the result set by the "CBname" column. The "SELECT" statement selects the "CBname" column from the "Shifts" table and the count of rows for each "CBname" group in the result set, which is renamed as "NoGuards" using the "AS" keyword. The "FROM" clause specifies that the "Guards" and "Shifts" tables are joined using the "GID" column as the common column. The "GROUP BY" clause groups the result set by the "CBname" column in the "Shifts" table, which means that the "COUNT" function will return the number of guards for each unique shift block in the "Shifts" • This SQL query selects the shift block name ("CBname"), cell ID ("CID"), and the number of inmates in each cell in the "Cell" table, by joining with the "Inmates" table using the common "CID" (cell ID) column, and grouping the result set by both "CBname" and "CID". The "SELECT" statement selects the "CBname", "CID", and the count of rows for each unique combination of "CBname" and "CID" group in the result set, which is renamed as "NoInmates" using the "AS" keyword. The "FROM" clause specifies that the "Inmates" and "Cell" tables are joined using the "CID" column as the common column. The "GROUP BY" clause groups the result set by both "CBname" and "CID" columns in the "Cell" table, which means that the "COUNT" function will return the number of inmates for each unique cell in each unique shift block in the "Cell" table.
1 union or intersection query	• This SQL query selects the "CID" column from the "Cell" table for cells where either the "CBname" column is equal to user specified cell block or the "MaxCap" column is equal to the user input. The "WHERE" clause filters the result set to include only the rows where the condition specified is true.
1 query to show that data that doesn't exist	• This SQL query selects the "CID" column from the "Cell" table for cells that do not have any inmates assigned to them in the "Inmates" table, by using a left join with the "Inmates" table and filtering out the rows where the "CID" value in the "Inmates" table is not null. • The "SELECT" statement selects the "CID" column from the "Cell" table. The "FROM" clause specifies the "Cell" table. The "LEFT JOIN" clause joins the "Inmates" table to the "Cell" table using the common "CID" (cell ID) column, returning all cells from the "Cell" table and any matching inmate records from the "Inmates" table, and NULL values for the

	inmate columns for any non-matching records. The "WHERE" clause filters the result set to only include the rows where the "CID" value in the "Inmates" table is NULL, which indicates that there are no inmates assigned to that cell.
1 Trigger	• This SQL query updates the "GType" column of the "Guards" table to 'Warden' for all guards who are currently labeled as 'Security officer' and have a "Clearance" value of 0, and there are no other guards currently labeled as 'Warden'. The "UPDATE" statement updates the "GType" column to 'Warden' for the specified guards. The "SET" clause specifies the new value for the "GType" column. The "WHERE" clause filters the rows to update only those where the guard's "GType" is currently 'Security officer' and their "Clearance" is 0. The "NOT EXISTS" subquery checks if there are no other guards in the table with a "GType" of 'Warden', using the "SELECT 1" statement to check for the existence of a row.

Nested Queries

SELECT *FROM Inmates

WHERE PID >

(Select AVG(PID) FROM Inmates WHERE Gender = 'M' and PConduct = 'Good');

This query is a SQL SELECT statement that selects all columns from the table “Inmates” where the value of “PID” is greater than the average value of “PID” for all male inmates with good conduct. The subquery (Select AVG(PID) FROM Inmates WHERE Gender = 'M' and PConduct = 'Good') calculates the average value of “PID” for all male inmates with good conduct, which is then used as a comparison value for the main query. The main query then selects all rows where “PID” is greater than this comparison value.

SELECT *FROM CellBlock WHERE NoCams > (Select AVG(NoCams) FROM CellBlock);

This query is a SQL SELECT statement that selects all columns from the table “CellBlock” where the value of “NoCams” is greater than the average value of “NoCams” for all cell blocks. The subquery (Select AVG(NoCams) FROM CellBlock) calculates the average value of “NoCams” for all cell blocks, which is then used as a comparison value for the main query. The main query then selects all rows where “NoCams” is greater than this comparison value

Main Operations

The main operations of the Prison Database system consist of the following:

1. Insert an Inmate
2. Move an Inmate
3. Count all Inmates in a certain Cell Block

The first one allows the user to input a new prisoner to the Database.

As for the second one, it can be used to move an inmate from his/her cell to another cell.

Finally, the third operation is designed to count the number of Inmates in a specified Cell Block.

Create Statements

```
CREATE TABLE CellBlock (
  CBname VARCHAR(50) PRIMARY KEY,
  CBType VARCHAR(50),
  NoCams INT
);

CREATE TABLE Cell (
  CID INT PRIMARY KEY,
  CBname VARCHAR(50),
  CellStatus VARCHAR(50),
  MaxCap INT,
  FOREIGN KEY (CBname) REFERENCES CellBlock(CBname)
);

CREATE TABLE Inmates (
  PFirst VARCHAR(50),
  PLast VARCHAR(50),
  PID INT PRIMARY KEY,
  Gender CHAR,
  PSentence INT,
  PGang VARCHAR(50),
  PConduct VARCHAR(50),
  Parole VARCHAR(3),
  CID INT,
  FOREIGN KEY (CID) REFERENCES Cell(CID)
);

CREATE TABLE Guards (
  GFirst VARCHAR(50),
  GLast VARCHAR(50),
  GID INT PRIMARY KEY,
  GType VARCHAR(50),
  Clearance INT NULL,
  Warden BIT NOT NULL
);

CREATE TABLE Shifts (
  Day VARCHAR(50),
  Time_ VARCHAR(50),
  GID INT,
  CBname VARCHAR(50),
  PRIMARY KEY(Day,Time_,GID,CBname),
  FOREIGN KEY (CBname) REFERENCES CellBlock(CBname),
  FOREIGN KEY (GID) REFERENCES Guards(GID)
);
```

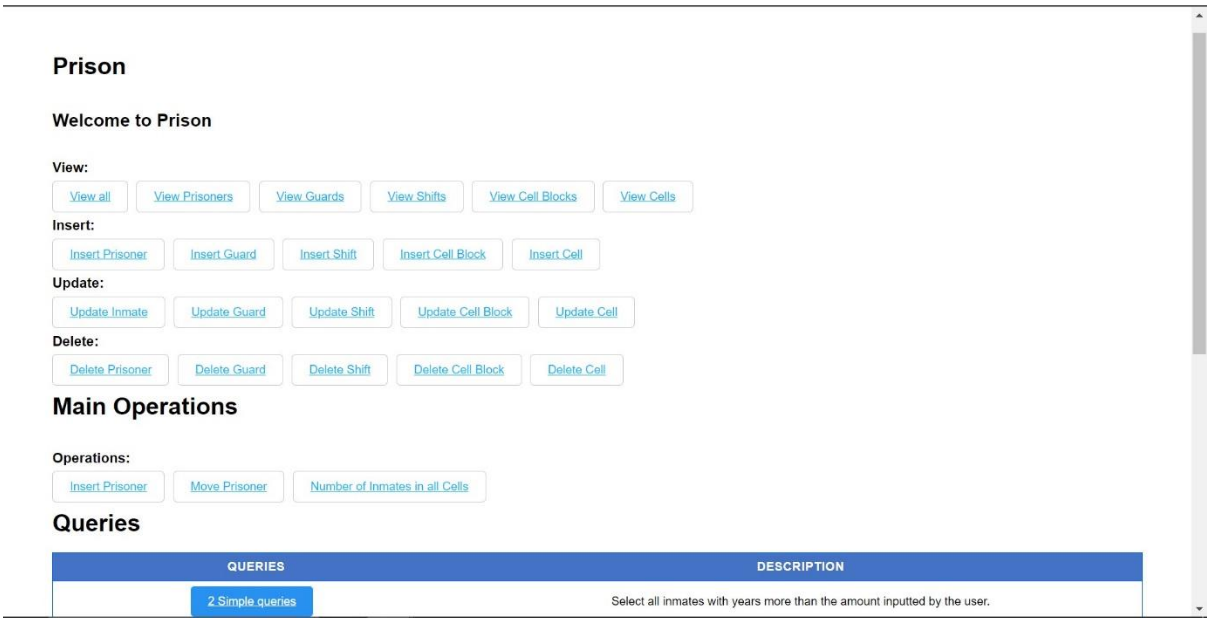
Insert Statements

```
INSERT INTO CellBlock (CBname, CBType, NoCams)
VALUES ('CB_A', 'Normal_M', 500);
INSERT INTO CellBlock (CBname, CBType, NoCams)
VALUES ('CB_B', 'Normal_M', 500);
INSERT INTO CellBlock (CBname, CBType, NoCams)
VALUES ('CB_C', 'Normal_F',);
INSERT INTO CellBlock (CBname, CBType, NoCams)
VALUES ('CB_SA', 'Solitary_M', 750);
INSERT INTO CellBlock (CBname, CBType, NoCams)
VALUES ('CB_SB', 'Solitary_M', 750);
INSERT INTO CellBlock (CBname, CBType, NoCams)
VALUES ('CB_SC', 'Solitary_F', 750);


INSERT INTO Cell (CBname, CellStatus, MaxCap)
VALUES ('CB_A', 'Empty', 2);
INSERT INTO Cell (CBname, CellStatus, MaxCap)
VALUES ('CB_B', 'Occupied', 4);
INSERT INTO Cell (CBname, CellStatus, MaxCap)
VALUES ('CB_C', 'Maintenance', 3);
INSERT INTO Cell (CBname, CellStatus, MaxCap)
VALUES ('CB_SA', 'Full', 1);
INSERT INTO Cell (CBname, CellStatus, MaxCap)
VALUES ('CB_SB', 'Full', 1);
INSERT INTO Cell (CBname, CellStatus, MaxCap)
VALUES ('CB_SA', 'Empty', 1);

INSERT INTO Inmates (PFirst, PLast, Gender, PSentence, PGang,
PConduct, Parole, CID)
VALUES ('Mostafa', 'Mohamed', 'M', 'Death Sentence', 'Cartel',
'Bad', 'No', 10000);
INSERT INTO Inmates (PFirst, PLast, Gender, PSentence, PGang,
PConduct, Parole, CID)
VALUES ('Khaled', 'Hany','M', '10', 'Cartel', 'Good', 'No', 10001);
INSERT INTO Inmates (PFirst, PLast, Gender, PSentence, PGang,
PConduct, Parole, CID)
VALUES ('Mariam', 'Samy', 'F', 3, NULL, 'Good', 'Yes', 10002);
INSERT INTO Inmates (PFirst, PLast, Gender, PSentence, PGang,
PConduct, Parole, CID)
VALUES ('Samar', 'Ahmed', 'F', '30 Years', NULL, 'Bad', 'No',
10003);
INSERT INTO Inmates (PFirst, PLast, Gender, PSentence, PGang,
PConduct, Parole, CID)
VALUES ('Ali', 'Ahmed','M', 'Death Sentence', 'Mafia', 'Good', 'No',
10004);
INSERT INTO Inmates (PFirst, PLast, Gender, PSentence, PGang,
PConduct, Parole, CID)
VALUES ('Hossam', 'Ashraf', 'M', 'Death Sentence', 'Irish Mob',
'Bad', 'No', 10005);
```

```
INSERT INTO Guards (GFirst, GLast, GType, Clearance, Warden)
VALUES ('Mohamed', 'Ahmed', 'Warden', 0, 1);
INSERT INTO Guards (GFirst, GLast, GType, Clearance, Warden)
VALUES ('John', 'Smith', 'Riot Unit', 1, 0);
INSERT INTO Guards (GFirst, GLast, GType, Clearance, Warden)
VALUES ('Hany', 'Hassanen', 'Watch Tower', 2, 0);
INSERT INTO Guards (GFirst, GLast, GType, Clearance, Warden)
VALUES ('Adel', 'Mohamed', 'CCTV Operator', 3, 0);
INSERT INTO Guards (GFirst, GLast, GType, Clearance, Warden)
VALUES ('Mark', 'Alaa', 'Security Officer', 4, 0);
INSERT INTO Guards (GFirst, GLast, GType, Clearance, Warden)
VALUES ('George', 'Washington', 'Security Officer', 4, 0);

INSERT INTO Shifts (Day, Time_, GID, CBname)
VALUES ('Wednesday', 'Day', 100, 'CB_A');
INSERT INTO Shifts (Day, Time_, GID, CBname)
VALUES ('Wednesday', 'Night', 101, 'CB_A');
INSERT INTO Shifts (Day, Time_, GID, CBname)
VALUES ('Tuesday', 'Day', 102, 'CB_A');
INSERT INTO Shifts (Day, Time_, GID, CBname)
VALUES ('Sunday', 'Night', 103, 'CB_B');
INSERT INTO Shifts (Day, Time_, GID, CBname)
VALUES ('Friday', 'Day', 104, 'CB_SC');
INSERT INTO Shifts (Day, Time_, GID, CBname)
VALUES ('Monday', 'Night', 105, 'CB_SB');
```

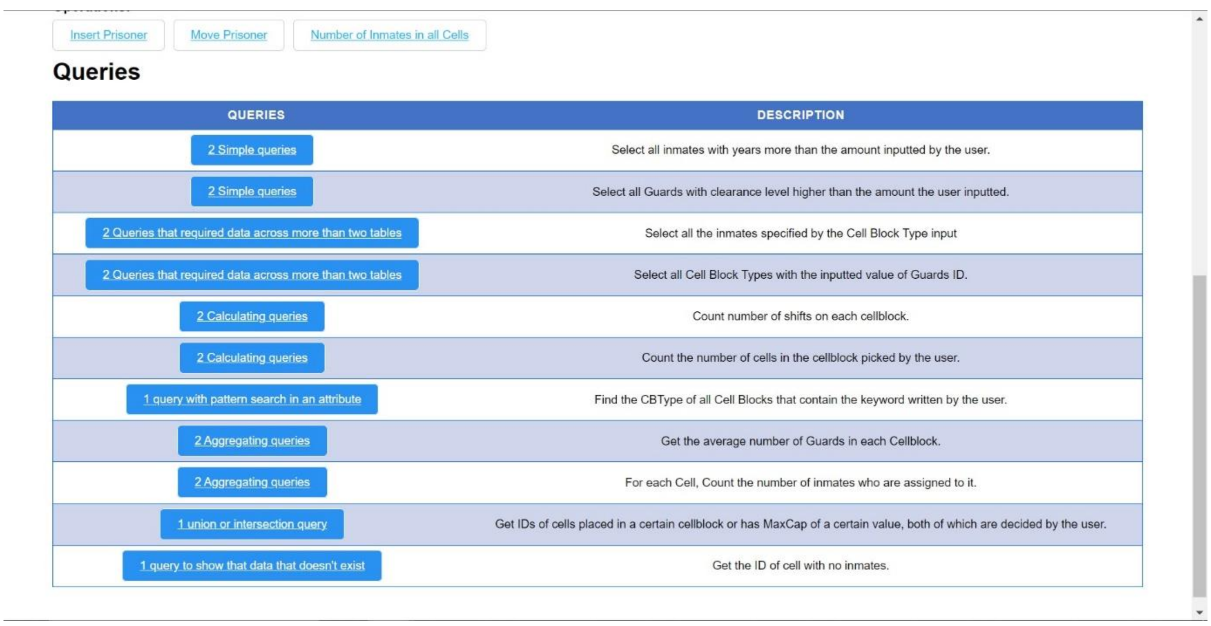
Interface screenshots

Main page, part 1:



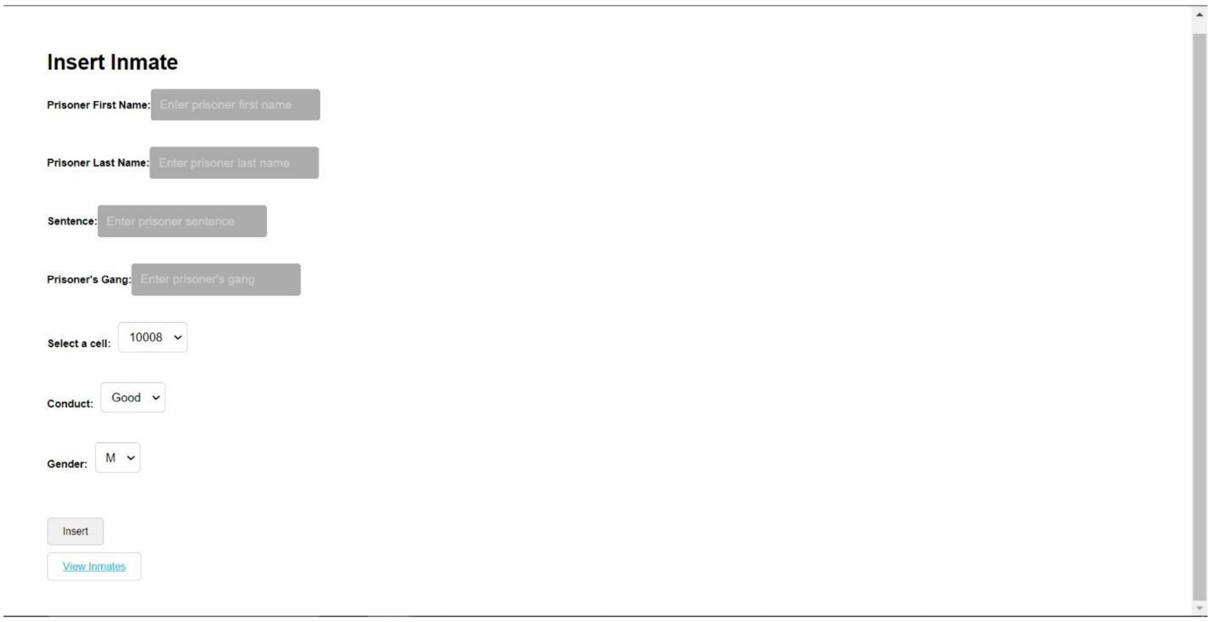
QUERIES	DESCRIPTION
2 Simple queries	Select all inmates with years more than the amount inputted by the user.

Main Page, part 2:



QUERIES	DESCRIPTION
2 Simple queries	Select all inmates with years more than the amount inputted by the user.
2 Simple queries	Select all Guards with clearance level higher than the amount the user inputted.
2 Queries that required data across more than two tables	Select all the inmates specified by the Cell Block Type input
2 Queries that required data across more than two tables	Select all Cell Block Types with the inputted value of Guards ID.
2 Calculating queries	Count number of shifts on each cellblock.
2 Calculating queries	Count the number of cells in the cellblock picked by the user.
1 query with pattern search in an attribute	Find the CBType of all Cell Blocks that contain the keyword written by the user.
2 Aggregating queries	Get the average number of Guards in each Cellblock.
2 Aggregating queries	For each Cell, Count the number of inmates who are assigned to it.
1 union or intersection query	Get IDs of cells placed in a certain cellblock or has MaxCap of a certain value, both of which are decided by the user.
1 query to show that data that doesn't exist	Get the ID of cell with no inmates.

Main Operation 1:



Insert Inmate

Prisoner First Name: Enter prisoner first name

Prisoner Last Name: Enter prisoner last name

Sentence: Enter prisoner sentence

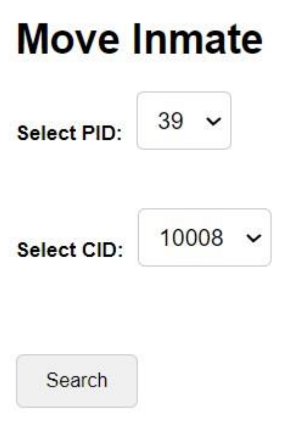
Prisoner's Gang: Enter prisoner's gang

Select a cell: 10008

Conduct: Good

Gender: M

Main Operation 2:

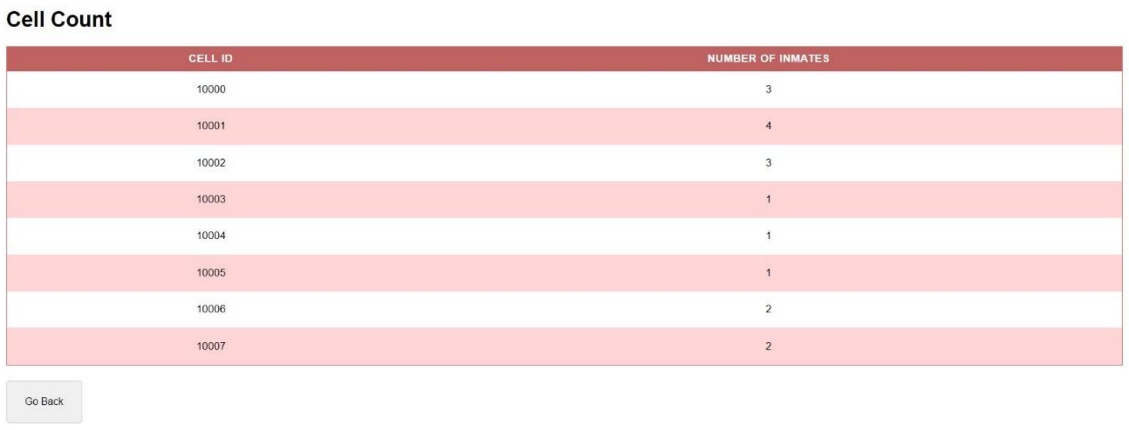


Move Inmate

Select PID: 39

Select CID: 10008

Main Operation 3:



Cell Count

CELL ID	NUMBER OF INMATES
10000	3
10001	4
10002	3
10003	1
10004	1
10005	1
10006	2
10007	2

Role of each member (Coding)

Mostafa Mohamed 212577: Insert Functionality, Remote access hosting (website)

Mohamed Ahmad 214618: Update Functionality, Database linking

Mohamed Abdel-Moneim 210210: Delete Functionality, HTML layout

All members: Java script Implementation