



## **INTRODUCTION TO DATABASE**

### **SEC – L**

**Faculty: JUENA AHMED NOSHIN**

### **Project**

## **PRISON MANAGEMENT SYSTEM**

### **Group-8**

Name	Id
Rahman, Nafiur	19-41457-3
Islam, Hasan Sanjary	19-39589-1
Hossan, Md. Riyaz	19-41432-3
Habib, Md Amir	19-41429-3
Ridoy, Tahmid Hassan	19-40307-1
Sakib, Safinul Ibrat	19-40298-1

# **INDEX**

INTRODUCTION.....	3
SCENARIO DESCRIPTION.....	4
ER DIAGRAM.....	5
NORMALIZATION.....	6
SCHEMA DIAGRAM.....	21
TABLE CREATION.....	22
DATA INSERTION.....	35
QUERY WRITING.....	45
RELATIONAL ALGEBRA.....	48
CONCLUSION.....	49

# **Introduction**

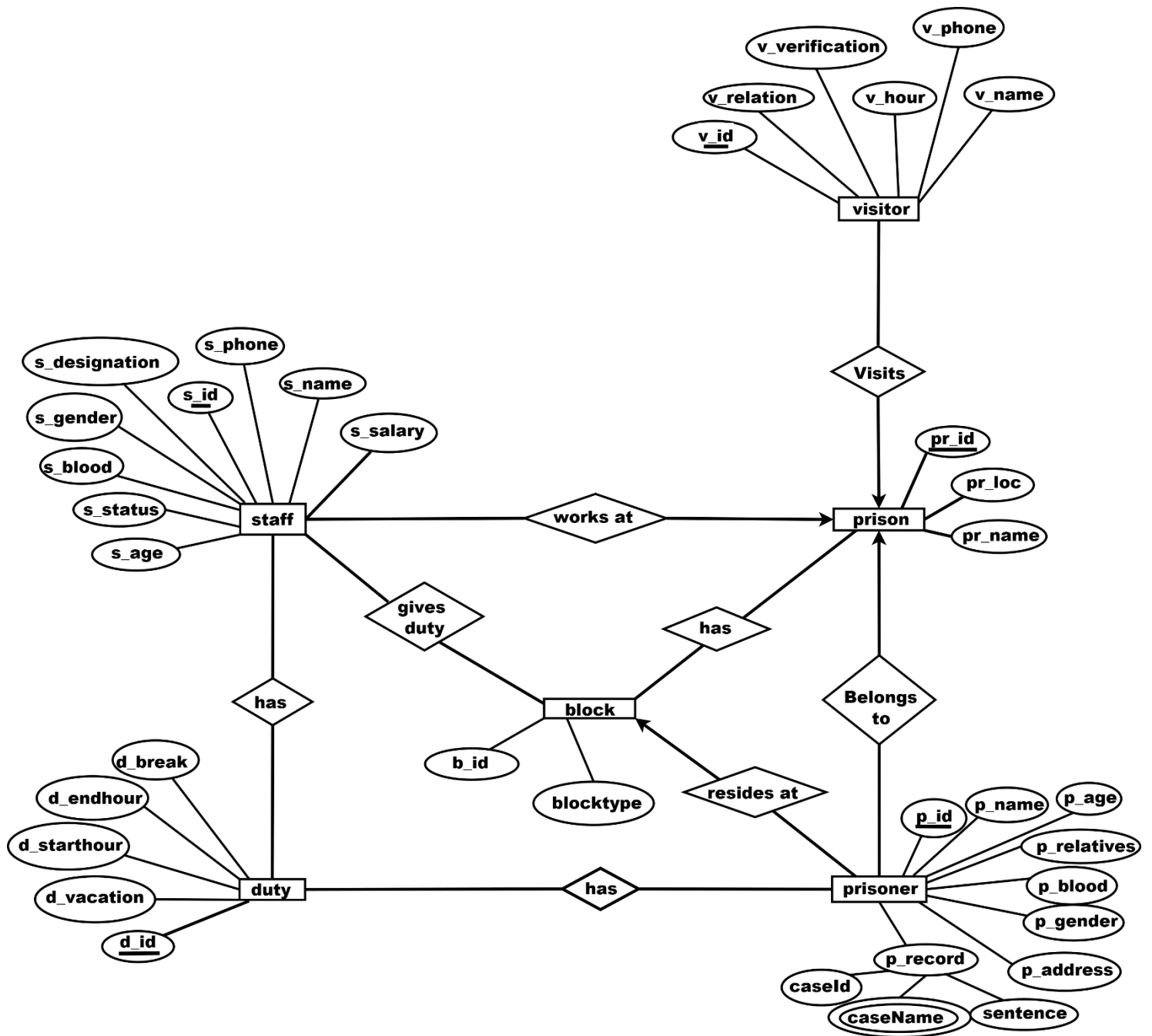
We are working on this prison management system with the hopes of making data more accessible and easily readable in prisons. Prisons are plagued with bad management system with hard to find data or sometimes data gets lost or misplaced, sometimes even leading to financial loss. We aim to solve that problem by providing an easy to navigate and easy to read management system which will show relevant data more efficiently.

Our project (Prison Management System) was created by the concept of DBMS.

## **Scenario**

In our Prison management system, many prisoners belong to one prison. A prisoner is identified by his or her prisoner id(p\_id). The system also stores name, age, blood group, gender, relatives and record. A prisoner record(p\_record) is multivalued attribute which stores case Id, case name and sentence. The prison is identified by id(pr\_id), name(pr\_name), and location(pr\_loc). Many prisons have many blocks. Inside the prison many prisoners reside in one block. Block has type(blocktype) and unique property of block id(b\_id). A prison also has many staff working in it. The staff is also identified by his or her id(s\_id). The system also stores age, gender, blood, designation, phone, salary, status. Many staff gives duty in many blocks. The staff and prisoner both have duties inside the prison. Many staff and many prisoners have many duties. The duty entity holds information like id(d\_id), start hour, end hour, break and vacation. Many visitors may visit one prison. The visitors have an id(v\_id) and the system stores information such as name, verification, hours of visit, phone number and relationship with the prisoner.

# Er Diagram



# **NORMALIZATION**

## **Staff works prison (Many to one)**

Unnormalized Form (UNF) :

Works (s\_age, s\_status, s\_blood, s\_gender, s\_designation, s\_id, s\_phone, s\_name, s\_salary, pr\_id, pr\_loc, pr\_name).

1NF ( 1<sup>st</sup> Normalized Form ) :

There is no multi valued attribute. Relation already in 1NF.

(s\_age, s\_status, s\_blood, s\_gender, s\_designation, s\_id, s\_phone, s\_name, s\_salary, pr\_id, pr\_loc, pr\_name).

2NF ( 2<sup>nd</sup> Normalized Form ) :

- s\_age, s\_status, s\_blood, s\_gender, s\_designation, s\_id, s\_phone, s\_name, s\_salary.
- pr\_id, pr\_loc, pr\_name

3NF (3<sup>rd</sup> Normalized Form):

There is no transitive dependency. Relation already in 3NF.

- s\_age, s\_status, s\_blood, s\_gender, s\_id, s\_phone, s\_name, s\_designation, s\_salary
- pr\_id, pr\_loc, pr\_name

### TABLE CREATION:

- s\_age, s\_status, s\_blood, s\_gender, s\_id, s\_phone, s\_name, s\_designation, s\_salary, **pr\_id**.
- pr\_id, pr\_loc, pr\_name.

## **Prisoner belongs to Prison (Many to One)**

### **Unnormalized Form (UNF) :**

Belongsto(p\_id, p\_name, p\_age, p\_relatives, p\_blood, p\_gender, p\_address, caseId, caseName, sentence, pr\_id, pr\_loc, pr\_name).

### **1NF ( 1<sup>st</sup> Normalized Form ) :**

caseName is a multi valued attribute.

- p\_id, p\_name, p\_age, p\_relatives, p\_blood, p\_gender, p\_address, caseId, caseName, sentence, pr\_id, pr\_loc, pr\_name.

### **2NF ( 2<sup>nd</sup> Normalized Form ) :**

- p\_id, p\_name, p\_age, p\_relatives, p\_blood, p\_gender, p\_address, caseId, caseName, sentence.
- pr\_id, pr\_loc, pr\_name.

### **3NF (3<sup>rd</sup> Normalized Form):**

- p\_id, p\_name, p\_age, p\_relatives, p\_blood, p\_gender, p\_address.
- caseId, caseName, sentence.
- pr\_id, pr\_loc, pr\_name.

### **TABLE CREATION:**

- p\_id, p\_name, p\_age, p\_relatives, p\_blood, p\_gender, p\_address, **A\_Id, pr\_id.**
- A\_Id, caseId, caseName, sentence.
- pr\_id, pr\_loc, pr\_name.



## Visitor visits Prison (Many to One)

### Unnormalized Form (UNF) :

Visits (v\_id, v\_relation, v\_verification, v\_hour, v\_phone, v\_name, pr\_id, pr\_loc, pr\_name).

### 1NF ( 1<sup>st</sup> Normalized Form ) :

There is no multi valued attribute. Relation already in 1NF.

- v\_id, v\_relation, v\_verification, v\_hour, v\_phone, v\_name, pr\_id, pr\_loc, pr\_name.

### 2NF ( 2<sup>nd</sup> Normalized Form ) :

- v\_id, v\_relation, v\_verification, v\_hour, v\_phone, v\_name.
- pr\_id, pr\_loc, pr\_name.

### 3NF (3<sup>rd</sup> Normalized Form):

There is no transitive dependency. Relation already in 3NF.

- v\_id, v\_relation, v\_verification, v\_hour, v\_phone, v\_name.
- pr\_id, pr\_loc, pr\_name.

### TABLE CREATION:

- v\_id, v\_relation, v\_verification, v\_hour, v\_phone, v\_name, **pr\_id**.
- pr\_id, pr\_loc, pr\_name.

## **Prison has Block (Many to Many)**

Unnormalized Form (UNF) :

Has (b\_id, blocktype, pr\_id, pr\_loc, pr\_name).

1NF ( 1<sup>st</sup> Normalized Form ) :

There is no multi valued attribute. Relation already in 1NF

- b\_id, blocktype, pr\_id, pr\_loc, pr\_name.

2NF ( 2<sup>nd</sup> Normalized Form ) :

- b\_id, blocktype.
- pr\_id, pr\_loc, pr\_name.

3NF (3<sup>rd</sup> Normalized Form):

There is no transitive dependency. Relation already in 3NF.

- b\_id, blocktype.
- pr\_id, pr\_loc, pr\_name.

TABLE CREATION:

- b\_id, blocktype.
- pr\_id, pr\_loc, pr\_name.
- **b\_id, pr\_id.**

## **Staff has Duty (Many to Many)**

### Unnormalized Form (UNF) :

Has (s\_age, s\_status, s\_blood, s\_gender, s\_designation, s\_id, s\_phone, s\_name, s\_salary, d\_id, d\_vacation, d\_starthour, d\_endhour, d\_break).

### 1NF ( 1<sup>st</sup> Normalized Form ) :

There is no multi valued attribute. Relation already in 1NF.

- s\_age, s\_status, s\_blood, s\_gender, s\_designation, s\_id, s\_phone, s\_name, s\_salary, d\_id, d\_vacation, d\_starthour, d\_endhour, d\_break.

### 2NF ( 2<sup>nd</sup> Normalized Form ) :

- s\_age, s\_status, s\_blood, s\_gender, s\_designation, s\_id, s\_phone, s\_name, s\_salary.
- d\_id, d\_vacation, d\_starthour, d\_endhour, d\_break.

### 3NF (3<sup>rd</sup> Normalized Form):

There is no transitive dependency. Relation already in 3NF.

- s\_age, s\_status, s\_blood, s\_gender, s\_id, s\_phone, s\_name, s\_designation, s\_salary.
- d\_id, d\_vacation, d\_starthour, d\_endhour, d\_break.

### TABLE CREATION:

- s\_age, s\_status, s\_blood, s\_gender, s\_id, s\_phone, s\_name, s\_id, s\_designation, s\_salary.
- d\_id, d\_vacation, d\_starthour, d\_endhour, d\_break.
- **s\_id, d\_id.**

## **Prisoner has Duty (Many to Many)**

### Unnormalized Form (UNF) :

Has (d\_id, d\_vacation, d\_starthour, d\_endhour, d\_break, p\_id, p\_name, p\_age, p\_relatives, p\_blood, p\_gender, p\_address, caseId, caseName, sentence).

### 1NF ( 1<sup>st</sup> Normalized Form ) :

caseName is a multi valued attribute.

- d\_id, d\_vacation, d\_starthour, d\_endhour, d\_break, p\_id, p\_name, p\_age, p\_relatives, p\_blood, p\_gender, p\_address, caseId, caseName, sentence.

### 2NF ( 2<sup>nd</sup> Normalized Form ) :

- d\_id, d\_vacation, d\_starthour, d\_endhour, d\_break.
- p\_id, p\_name, p\_age, p\_relatives, p\_blood, p\_gender, p\_address, caseId, caseName, sentence.

### 3NF (3<sup>rd</sup> Normalized Form):

- d\_id, d\_vacation, d\_starthour, d\_endhour, d\_break.
- p\_id, p\_name, p\_age, p\_relatives, p\_blood, p\_gender, p\_address.
- caseId, caseName, sentence.

### TABLE CREATION:

- d\_id, d\_vacation, d\_starthour, d\_endhour, d\_break.
- p\_id, p\_name, p\_age, p\_relatives, p\_blood, p\_gender, p\_address, **A\_Id**.
- A\_Id, caseId, caseName, sentence.
- **d\_id, p\_id**.

## **Staff gives duty in Blocks (Many to Many)**

### **Unnormalized Form (UNF) :**

Givesduty (s\_age, s\_status, s\_blood, s\_gender, s\_designation, s\_id, s\_phone, s\_name, s\_salary, b\_id, blocktype).

### **1NF ( 1<sup>st</sup> Normalized Form ) :**

There is no multi valued attribute. Relation already in 1NF.

- s\_age, s\_status, s\_blood, s\_gender, s\_designation, s\_id, s\_phone, s\_name, s\_salary, b\_id, blocktype.

### **2NF ( 2<sup>nd</sup> Normalized Form ) :**

- s\_age, s\_status, s\_blood, s\_gender, s\_designation, s\_id, s\_phone, s\_name, s\_salary.
- b\_id, blocktype.

### **3NF (3<sup>rd</sup> Normalized Form):**

There is no transitive dependency. Relation already in **3NF**.

- s\_age, s\_status, s\_blood, s\_gender, s\_id, s\_phone, s\_name, s\_designation, s\_salary.
- b\_id, blocktype.

### TABLE CREATION:

- s\_age, s\_status, s\_blood, s\_gender, s\_id, s\_phone, s\_name, s\_id, s\_designation, s\_salary.
- b\_id, blocktype.
- **s\_id, b\_id.**



## **Prisoner resides at Block (Many to one)**

### Unnormalized Form (UNF) :

Reside (p\_id, p\_name, p\_age, p\_relatives, p\_blood, p\_gender, p\_address, caseId, caseName, sentence, b\_id, blocktype).

### 1NF ( 1<sup>st</sup> Normalized Form ) :

caseName is a multi valued attribute.

- p\_id, p\_name, p\_age, p\_relatives, p\_blood, p\_gender, p\_address, caseId, caseName, sentence, b\_id, blocktype.

### 2NF ( 2<sup>nd</sup> Normalized Form ) :

- p\_id, p\_name, p\_age, p\_relatives, p\_blood, p\_gender, p\_address, caseId, caseName, sentence.
- b\_id, blocktype.

### 3NF (3<sup>rd</sup> Normalized Form):

- p\_id, p\_name, p\_age, p\_relatives, p\_blood, p\_gender, p\_address.
- caseId, caseName, sentence.
- b\_id, blocktype.

### TABLE CREATION:

- p\_id, p\_name, p\_age, p\_relatives, p\_blood, p\_gender, p\_address, **A\_id, b\_id**.
- A\_id, caseId, caseName, sentence.
- b\_id, blocktype.

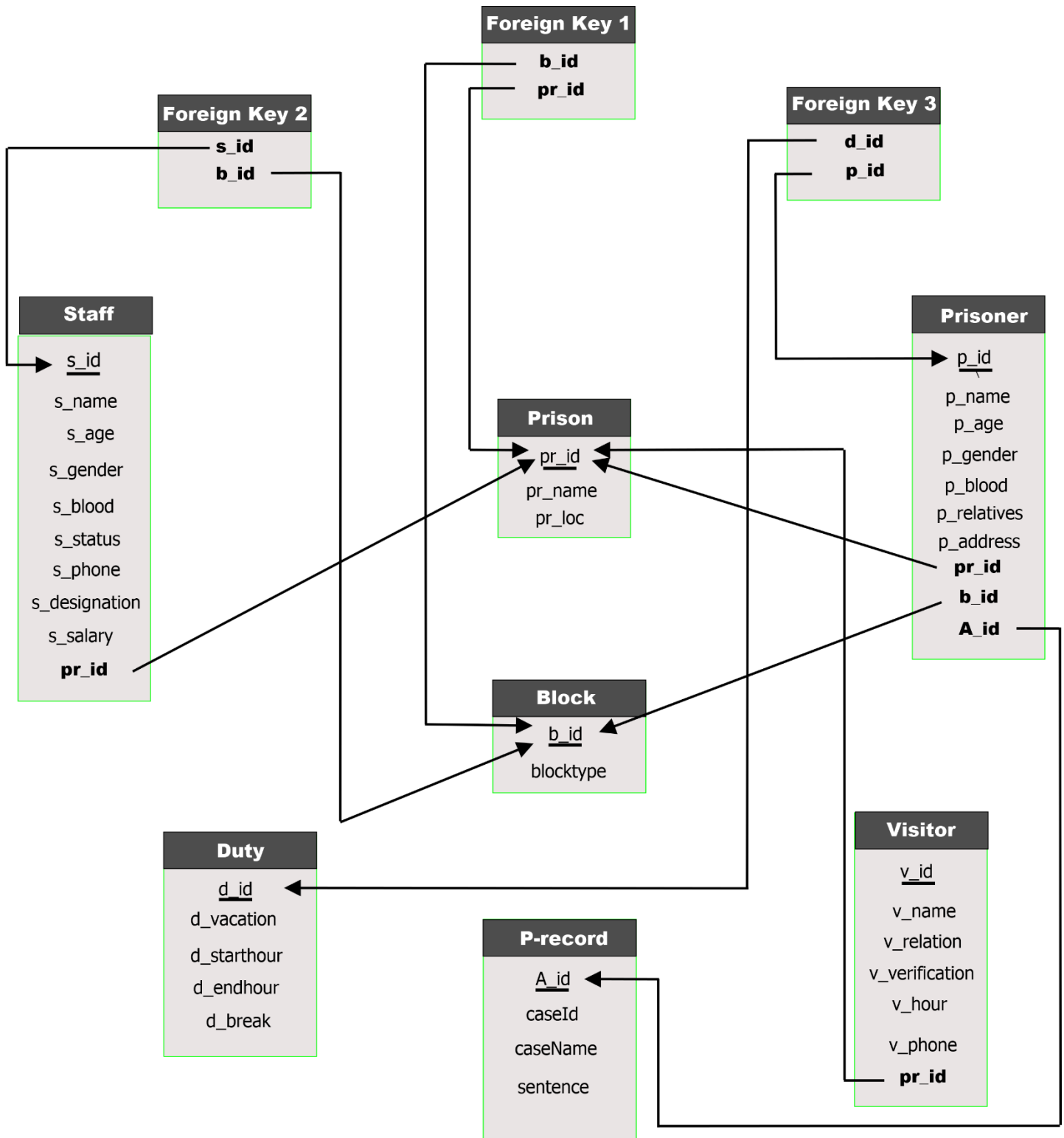
## TEMPORARY TABLES:

- s\_id, s\_name, s\_age, s\_gender, s\_blood, s\_status, s\_phone, , s\_designation, s\_salary, **pr\_id**.
- ~~pr\_id, pr\_loc, pr\_name.~~
- p\_id, p\_name, p\_age, p\_gender, p\_blood, p\_relatives, p\_address, **A\_id**, **pr\_id**.
- ~~A\_id, caseId, caseName, sentence.~~
- ~~pr\_id, pr\_loc, pr\_name.~~
- v\_id, v\_relation, v\_verification, v\_hour, v\_phone, v\_name, **pr\_id**.
- ~~pr\_id, pr\_loc, pr\_name.~~
- ~~b\_id, blocktype.~~
- pr\_id, pr\_name, pr\_loc.
- **b\_id, pr\_id**.
- ~~s\_age, s\_status, s\_blood, s\_gender, s\_id, s\_phone, s\_name, s\_id, s\_designation, s\_salary.~~
- ~~d\_id, d\_vacation, d\_starthour, d\_endhour, d\_break.~~
- ~~s\_id, d\_id.~~
- d\_id, d\_vacation, d\_starthour, d\_endhour, d\_break.
- ~~p\_id, p\_name, p\_age, p\_relatives, p\_blood, p\_gender, p\_address, A\_id.~~
- ~~A\_id, caseId, caseName, sentence.~~
- **d\_id, p\_id**.
- ~~s\_age, s\_status, s\_blood, s\_gender, s\_id, s\_phone, s\_name, s\_id, s\_designation, s\_salary.~~
- ~~b\_id, blocktype.~~
- **s\_id, b\_id**.
- p\_id, p\_name, p\_age, p\_gender, p\_blood, p\_relatives, p\_address, **A\_id**, **b\_id**.
- A\_id, caseId, caseName, sentence.
- b\_id, blocktype.

### **FINAL TABLE:**

- s\_id, s\_name, s\_age, s\_gender, s\_blood, s\_status, s\_phone, , s\_designation, s\_salary, **pr\_id**.
- p\_id, p\_name, p\_age, p\_gender, p\_blood, p\_relatives, p\_address, **A\_id, pr\_id**.
- v\_id, v\_name, v\_relation, v\_verification, v\_hour, v\_phone, **pr\_id**.
- pr\_id, pr\_name, pr\_loc.
- **b\_id, pr\_id**.
- d\_id, d\_vacation, d\_starthour, d\_endhour, d\_break.
- **d\_id, p\_id**.
- **s\_id, b\_id**.
- p\_id, p\_name, p\_age, p\_gender, p\_blood, p\_relatives, p\_address, **A\_id, b\_id**.
- A\_id, caseId, caseName, sentence.
- b\_id, blocktype.

## SCHEMA DIAGRAM



# Table Creation

```
1.CREATE TABLE Staff(
    s_id number(10) PRIMARY KEY,
    s_name varchar2(20),
    s_age number(10),
    s_gender varchar2(20),
    s_blood varchar2(20),
    s_status varchar2(20),
    s_phone number(20),
    s_designation varchar2(20),
    s_salary number(20),
    pr_id number(10)
);
```

Results Explain Describe **Saved SQL** History

Object Type **TABLE** Object **STAFF**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<u>STAFF</u>	<u>S_ID</u>	Number	-	10	0	1	-	-	-
	<u>S_NAME</u>	Varchar2	20	-	-	-	✓	-	-
	<u>S_AGE</u>	Number	-	10	0	-	✓	-	-
	<u>S_GENDER</u>	Varchar2	20	-	-	-	✓	-	-
	<u>S_BLOOD</u>	Varchar2	20	-	-	-	✓	-	-
	<u>S_STATUS</u>	Varchar2	20	-	-	-	✓	-	-
	<u>S_PHONE</u>	Number	-	20	0	-	✓	-	-
	<u>S_DESIGNATION</u>	Varchar2	20	-	-	-	✓	-	-
	<u>S_SALARY</u>	Number	-	20	0	-	✓	-	-
	<u>PR_ID</u>	Number	-	10	0	-	✓	-	-
									1 - 10

```

2. CREATE TABLE Prisoner(
    p_id number(10) PRIMARY KEY,
    p_name varchar2(20),
    p_age number(10),
    p_gender varchar2(20),
    p_blood varchar2(20),
    p_relatives varchar2(20),
    p_address varchar2(20),
    pr_id number(10),
    b_id number(10),
    A_id number(10)
);

```

Results Explain Describe **Saved SQL** History

Object Type TABLE Object PRISONER

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<u>PRISONER</u>	<u>P_ID</u>	Number	-	10	0	1	-	-	-
	<u>P_NAME</u>	Varchar2	20	-	-	-	✓	-	-
	<u>P_AGE</u>	Number	-	10	0	-	✓	-	-
	<u>P_GENDER</u>	Varchar2	20	-	-	-	✓	-	-
	<u>P_BLOOD</u>	Varchar2	20	-	-	-	✓	-	-
	<u>P_RELATIVES</u>	Varchar2	20	-	-	-	✓	-	-
	<u>P_ADDRESS</u>	Varchar2	20	-	-	-	✓	-	-
	<u>PR_ID</u>	Number	-	10	0	-	✓	-	-
	<u>B_ID</u>	Number	-	10	0	-	✓	-	-
	<u>A_ID</u>	Number	-	10	0	-	✓	-	-
1 - 10									

```

3. CREATE TABLE Visitor(
    v_id number(10),
    v_name varchar2(20),
    v_relation varchar2(20),
    v_verification number(20),
    v_hour varchar2(20),
    v_phone number(20),
    pr_id number(10)
);

```

Results Explain Describe Saved SQL History

Object Type TABLE Object VISITOR

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<a href="#">VISITOR</a>	<a href="#">V ID</a>	Number	-	10	0	-	✓	-	-
	<a href="#">V NAME</a>	Varchar2	20	-	-	-	✓	-	-
	<a href="#">V RELATION</a>	Varchar2	20	-	-	-	✓	-	-
	<a href="#">V VERIFICATION</a>	Number	-	20	0	-	✓	-	-
	<a href="#">V HOUR</a>	Varchar2	20	-	-	-	✓	-	-
	<a href="#">V PHONE</a>	Number	-	20	0	-	✓	-	-
	<a href="#">PR ID</a>	Number	-	10	0	-	✓	-	-

1 - 7



```

4. CREATE TABLE P_record(
    A_id number(10) PRIMARY KEY,
    caseId number(10),
    caseName varchar2(20),
    sentence varchar2(20)
);

```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **P\_RECORD**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
P_RECORD	<u>A_ID</u>	Number	-	10	0	1	-	-	-
	<u>CASEID</u>	Number	-	10	0	-	✓	-	-
	<u>CASENAME</u>	Varchar2	20	-	-	-	✓	-	-
	<u>SENTENCE</u>	Varchar2	20	-	-	-	✓	-	-
									1 - 4

```

5. CREATE TABLE Duty(
    d_id number(10) PRIMARY KEY,
    d_vacation varchar2(20),
    d_starthour varchar2(20),
    d_endhour varchar2(20),
    d_break varchar2(20)
);

```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **DUTY**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<u>DUTY</u>	<u>D_ID</u>	Number	-	10	0	1	-	-	-
	<u>D_VACATION</u>	Varchar2	20	-	-	-	✓	-	-
	<u>D_STARTHOUR</u>	Varchar2	20	-	-	-	✓	-	-
	<u>D_ENDHOUR</u>	Varchar2	20	-	-	-	✓	-	-
	<u>D_BREAK</u>	Varchar2	20	-	-	-	✓	-	-

1 - 5

```

6. CREATE TABLE Prison(
    pr_id number(10) PRIMARY KEY,
    pr_name varchar2(20),
    pr_loc varchar2(20)
);

```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **PRISON**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<u>PRISON</u>	<u>PR_ID</u>	Number	-	10	0	1	-	-	-
	<u>PR_NAME</u>	Varchar2	20	-	-	-	✓	-	-
	<u>PR_LOC</u>	Varchar2	20	-	-	-	✓	-	-
									1 - 3

```

7. CREATE TABLE Block(
    b_id number(10) PRIMARY KEY,
    blocktype varchar2(20)
);

```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **BLOCK**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<u>BLOCK</u>	<u>B_ID</u>	Number	-	10	0	1	-	-	-
	<u>BLOCKTYPE</u>	Varchar2	20	-	-	-	✓	-	-
1 - 2									

```

8. CREATE TABLE ForeignKey1(
    b_id number(10),
    pr_id number(10)
);

```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **FOREIGNKEY1**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
FOREIGNKEY1	<u>B_ID</u>	Number	-	10	0	-	✓	-	-
	<u>PR_ID</u>	Number	-	10	0	-	✓	-	-
1 - 2									

```

9. CREATE TABLE ForeignKey2(
    s_id number(10),
    b_id number(10)
);

```

Results Explain Describe Saved SQL History

Object Type TABLE Object FOREIGNKEY2

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
FOREIGNKEY2	S_ID	Number	-	10	0	-	✓	-	-
	B_ID	Number	-	10	0	-	✓	-	-
1 - 2									

```

10. CREATE TABLE ForeignKey3(
    d_id number(10),
    p_id number(10)
);

```

Results Explain Describe Saved SQL History

Object Type TABLE Object FOREIGNKEY3

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
FOREIGNKEY3	D_ID	Number	-	10	0	-	✓	-	-
	P_ID	Number	-	10	0	-	✓	-	-
									1 - 2

## **Constraints**

- 1.ALTER TABLE Staff ADD CONSTRAINT FK1 FOREIGN KEY(pr\_id) REFERENCES Prison(pr\_id);
- 2.ALTER TABLE Visitor ADD CONSTRAINT FK2 FOREIGN KEY(pr\_id) REFERENCES Prison(pr\_id);
- 3.ALTER TABLE Prisoner ADD CONSTRAINT FK3 FOREIGN KEY(pr\_id) REFERENCES Prison(pr\_id);
- 4.ALTER TABLE Prisoner ADD CONSTRAINT FK4 FOREIGN KEY(b\_id) REFERENCES Block(b\_id);
- 5.ALTER TABLE Prisoner ADD CONSTRAINT FK5 FOREIGN KEY(A\_id) REFERENCES P\_record(A\_id);
- 6.ALTER TABLE ForeignKey1 ADD CONSTRAINT FK6 FOREIGN KEY(b\_id) REFERENCES Block(b\_id);
- 7.ALTER TABLE ForeignKey1 ADD CONSTRAINT FK7 FOREIGN KEY(pr\_id) REFERENCES Prison(pr\_id);
- 8.ALTER TABLE ForeignKey2 ADD CONSTRAINT FK8 FOREIGN KEY(s\_id) REFERENCES Staff(s\_id);
- 9.ALTER TABLE ForeignKey2 ADD CONSTRAINT FK9 FOREIGN KEY(b\_id) REFERENCES Block(b\_id);
- 10.ALTER TABLE ForeignKey3 ADD CONSTRAINT FK10 FOREIGN KEY(d\_id) REFERENCES Duty(d\_id);
- 11.ALTER TABLE ForeignKey3 ADD CONSTRAINT FK11 FOREIGN KEY(p\_id) REFERENCES Prisoner(p\_id);



## Create User

☒ Autocommit   Display 10 ▼

CREATE USER Lucifer  
IDENTIFIED BY prisonbreak;

Results   Explain   Describe   Saved SQL   History

User created.

0.02 seconds

## Grant unlimited tablespace

☒ Autocommit   Display 10 ▼

GRANT UNLIMITED TABLESPACE TO Lucifer

Results   Explain   Describe   Saved SQL   History

Statement processed.

0.00 seconds

## Create Role

☒ Autocommit   Display 10 ▾

CREATE ROLE PrisonHead;

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

Role created.

0.00 seconds

## Grant Privileges

☒ Autocommit   Display 10 ▾

GRANT CREATE TABLE, CREATE VIEW , CREATE SEQUENCE  
TO PrisonHead;

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

Statement processed.

0.18 seconds

## Assign Roles

☒ Autocommit   Display 10 ▾

GRANT PrisonHead TO Lucifer;

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

Statement processed.

0.18 seconds

## Data Insertion

### Prison Table:

- INSERT INTO Prison  
VALUES(101,'St. Clair Prison','St. Clair');
- INSERT INTO Prison  
VALUES(102,'Bullock Prison','Bullock');
- INSERT INTO Prison  
VALUES(103,'Elmore Prison','Elmore');
- INSERT INTO Prison  
VALUES(104,'Limestone Prison','Madison');
- INSERT INTO Prison  
VALUES(105,'Ventress Prison','Barbour');

**Results** Explain Describe Saved SQL History

PR_ID	PR_NAME	PR_LOC
101	St. Clair Prison	St. Clair
102	Bullock Prison	Bullock
103	Elmore Prison	Elmore
104	Limestone Prison	Madison
105	Ventress Prison	Barbour

### Duty Table:

- INSERT INTO Duty  
VALUES(201,'In Vacation','8-AM','6-PM','1PM to 1:30PM');
- INSERT INTO Duty  
VALUES(202,'0 Days Available','8-AM','6-PM','1PM to 1:30PM');
- INSERT INTO Duty  
VALUES(203,'7 Days Available','8-AM','6-PM','1PM to 1:30PM');
- INSERT INTO Duty  
VALUES(204,'4 Days Available','8-PM','6-AM','2AM to 2:30AM');
- INSERT INTO Duty  
VALUES(205,'0 Days Available','8-PM','6-AM','2AM to 2:30AM');

**Results** Explain Describe Saved SQL History

D_ID	D_VACATION	D_STARTHOUR	D_ENDHOUR	D_BREAK
201	In Vacation	8-AM	6-PM	1PM to 1:30PM
202	0 Days Available	8-AM	6-PM	1PM to 1:30PM
203	7 Days Available	8-AM	6-PM	1PM to 1:30PM
204	4 Days Available	8-PM	6-AM	2AM to 2:30AM
205	0 Days Available	8-PM	6-AM	2AM to 2:30AM

### **Block Table:**

- INSERT INTO Block  
VALUES(1001,'Block-A(Male)');
- INSERT INTO Block  
VALUES(1002,'Block-B(Female)');
- INSERT INTO Block  
VALUES(1003,'Block-C(Male)');
- INSERT INTO Block  
VALUES(1004,'Block-D(Female)');
- INSERT INTO Block  
VALUES(1005,'Block-E(Male)');

**Results** Explain Describe Saved SQL History

B_ID	BLOCKTYPE
1001	Block-A(Male)
1002	Block-B(Female)
1003	Block-C(Male)
1004	Block-D(Female)
1005	Block-E(Male)

### **P-record Table:**

- INSERT INTO P\_record  
VALUES(111, 12001,'Robbery','3 Years');
- INSERT INTO P\_record  
VALUES(112, 12002,'Rape','9 Years');
- INSERT INTO P\_record  
VALUES(113, 12003,'Homicide','Life sentence');
- INSERT INTO P\_record  
VALUES(114, 12004,'Drug Possession','8 Months');
- INSERT INTO P\_record  
VALUES(115, 12005,'Cyberbullying','6 Months');

**Results** Explain Describe Saved SQL History

A_ID	CASEID	CASENAME	SENTENCE
111	12001	Robbery	3 Years
112	12002	Rape	9 Years
113	12003	Homicide	Life sentence
114	12004	Drug Possession	8 Months
115	12005	Cyberbullying	6 Months

### Visitor Table:

- INSERT INTO Visitor  
VALUES(1010,'Jack Quaid','Husband',3412,'10:00AM-10:20AM',356741256,101);
- INSERT INTO Visitor  
VALUES(1011,'Shantel VanSanten','Brother',3413,'10:00AM-10:20AM',2459631258,101);
- INSERT INTO Visitor  
VALUES(1012,'Timothée Chalamet','Sister',3414,'12:00PM-12:20PM',7965482358,101);
- INSERT INTO Visitor  
VALUES(1013,'Emily Alyn Lind','Mother',3415,'10:00AM-10:20AM',6542549534,101);
- INSERT INTO Visitor  
VALUES(1014,'Bella Thorne','Wife',3416,'12:00PM-12:20PM',8546219547,101);

**Results** Explain Describe Saved SQL History

V_ID	V_NAME	V_RELATION	V_VERIFICATION	V_HOUR	V_PHONE	PR_ID
1010	Jack Quaid	Husband	3412	10:00AM-10:20AM	356741256	101
1011	Shantel VanSanten	Brother	3413	10:00AM-10:20AM	2459631258	101
1012	Timothée Chalamet	Sister	3414	12:00PM-12:20PM	7965482358	101
1013	Emily Alyn Lind	Mother	3415	10:00AM-10:20AM	6542549534	101
1014	Bella Thorne	Wife	3416	12:00PM-12:20PM	8546219547	101

### Staff Table:

- INSERT INTO Staff  
VALUES(1101, 'Zakaria  
Brooks',35,'Male','O+','ACTIVE',1714445555,'Deputy Warden',3000,101);
- INSERT INTO Staff  
VALUES(1102, 'Joshua  
Nicole',24,'Female','A+','ACTIVE',2223331111,'Probation  
Officer',2500,101);
- INSERT INTO Staff  
VALUES(1103, 'Sasha  
Grey',25,'Female','B+','ACTIVE',4455552233,'Treatment  
Specialist',1500,101);
- INSERT INTO Staff  
VALUES(1104, 'Jon Rogers',30,'Male','O-','ACTIVE',6666887799,'Prison  
Guard',1000,101);
- INSERT INTO Staff  
VALUES(1105, 'Alex Cox',32,'Male','B+','INACTIVE',9999666777,'Safety  
Specialist',1800,101);

Results Explain Describe Saved SQL History

S_ID	S_NAME	S_AGE	S_GENDER	S_BLOOD	S_STATUS	S_PHONE	S_DESIGNATION	S_SALARY	PR_ID
1101	Zakaria Brooks	35	Male	O+	ACTIVE	1714445555	Deputy Warden	3000	101
1102	Joshua Nicole	24	Female	A+	ACTIVE	2223331111	Probation Officer	2500	101
1103	Sasha Grey	25	Female	B+	ACTIVE	4455552233	Treatment Specialist	1500	101
1104	Jon Rogers	30	Male	O-	ACTIVE	6666887799	Prison Guard	1000	101
1105	Alex Cox	32	Male	B+	INACTIVE	9999666777	Safety Specialist	1800	101



### Prisoner Table:

- INSERT INTO Prisoner  
VALUES(1,'Jim Jones',27,'Male','O+','Bella  
Thorne','Borger,Texas',101,1001,111);
- INSERT INTO Prisoner  
VALUES(2,'Albert L. Bates',30,'Male','A+','Emily Alyn  
Lind','Granada,California',101,1001,112);
- INSERT INTO Prisoner  
VALUES(3,'Charlotte Corday',26,'Female','O-','Shantel  
VanSanten','Tishomingo,Oklahoma',101,1002,113);
- INSERT INTO Prisoner  
VALUES(4,'Samantha Lewthwaite',29,'Female','B+','Jack  
Quaid','Pikeville,Kentucky',101,1004,114);
- INSERT INTO Prisoner  
VALUES(5,'Arthur Barker',32,'Male','A+','Timothée  
Chalamet','Musella,Georgia',101,1005,115);

Results Explain Describe Saved SQL History

P_ID	P_NAME	P_AGE	P_GENDER	P_BLOOD	P_RELATIVES	P_ADDRESS	PR_ID	B_ID	A_ID
1	Jim Jones	27	Male	O+	Bella Thorne	Borger,Texas	101	1001	111
2	Albert L. Bates	30	Male	A+	Emily Alyn Lind	Granada,California	101	1001	112
3	Charlotte Corday	26	Female	O-	Shantel VanSanten	Tishomingo,Oklahoma	101	1002	113
4	Samantha Lewthwaite	29	Female	B+	Jack Quaid	Pikeville,Kentucky	101	1004	114
5	Arthur Barker	32	Male	A+	Timothée Chalamet	Musella,Georgia	101	1005	115

### **Foreign Key 1 Table:**

- INSERT INTO Foreignkey1  
VALUES(1001,101);
- INSERT INTO Foreignkey1  
VALUES(1002,102);
- INSERT INTO Foreignkey1  
VALUES(1003,103);
- INSERT INTO Foreignkey1  
VALUES(1004,104);
- INSERT INTO Foreignkey1  
VALUES(1005,105);

**Results** Explain Describe Saved SQL History

B_ID	PR_ID
1001	101
1002	102
1003	103
1004	104
1005	105

### Foreign Key 2 Table:

- INSERT INTO Foreignkey2  
VALUES(1101,1001);
- INSERT INTO Foreignkey2  
VALUES(1102,1002);
- INSERT INTO Foreignkey2  
VALUES(1103,1003);
- INSERT INTO Foreignkey2  
VALUES(1104,1004);
- INSERT INTO Foreignkey2  
VALUES(1105,1005);

**Results** Explain Describe Saved SQL History

S_ID	B_ID
1101	1001
1102	1002
1103	1003
1104	1004
1105	1005

### Foreign Key 3 Table:

- INSERT INTO Foreignkey3  
VALUES(201,1);
- INSERT INTO Foreignkey3  
VALUES(202,2);
- INSERT INTO Foreignkey3  
VALUES(203,3);
- INSERT INTO Foreignkey3  
VALUES(204,4);
- INSERT INTO Foreignkey3  
VALUES(205,5);

**Results** Explain Describe Saved SQL History

D_ID	P_ID
201	1
202	2
203	3
204	4
205	5

# Query Writing

## Subquery:

**Question-1:** Display the prisoner's name, id, gender who have "A+" blood group.

**Answer:** SELECT P\_name, P\_id, P\_gender FROM Prisoner

Where P\_blood = 'A+';

<input checked="" type="checkbox"/> Autocommit	Display	10	▼
SELECT P_name, P_id, P_gender FROM Prisoner Where P_blood = 'A+';			
Results Explain Describe Saved SQL History			
P_NAME	P_ID	P_GENDER	
Albert L. Bates	2	Male	
Arthur Barker	5	Male	
2 rows returned in 0.05 seconds			
<a href="#">CSV Export</a>			

**Question-2:** Display Staff's id, name, gender, salary and designation of those who have salary higher than 'Alex cox'.

**Answer:** SELECT S\_id, S\_name, S\_gender, S\_salary, S\_designation FROM Staff  
WHERE S\_salary > (SELECT S\_salary FROM Staff WHERE S\_name = 'Alex Cox');

☒ Autocommit

Display

10

▼

SELECT S\_id, S\_name, S\_gender, S\_salary, S\_designation FROM Staff |  
WHERE S\_salary > (SELECT S\_salary FROM Staff WHERE S\_name = 'Alex Cox');

Results

Explain

Describe

Saved SQL

History

S_ID	S_NAME	S_GENDER	S_SALARY	S_DESIGNATION
1101	Zakaria Brooks	Male	3000	Deputy Warden
1102	Joshua Nicole	Female	2500	Probation Officer

2 rows returned in 0.00 seconds

[CSV Export](#)

## Joining:

**Question-1:** Display prisoner's id, name, gender, casename and sentence of the prisoners.

**Answer:** SELECT P\_id, P\_name, P\_gender, casename, sentence FROM Prisoner, P\_record

WHERE PRISONER.A\_id=P\_record.A\_id;

☒ Autocommit

Display

10

▼

SELECT P\_id, P\_name, P\_gender, casename, sentence FROM Prisoner, P\_record  
WHERE PRISONER.A\_id=P\_record.A\_id;

Results

Explain

Describe

Saved SQL

History

P_ID	P_NAME	P_GENDER	CASENAME	SENTENCE
1	Jim Jones	Male	Robbery	3 Years
2	Albert L. Bates	Male	Rape	9 Years
3	Charlotte Corday	Female	Homicide	Life sentence
4	Samantha Lewthwaite	Female	Drug Possession	8 Months
5	Arthur Barker	Male	Cyberbullying	6 Months

5 rows returned in 0.02 seconds

CSV Export

**QUESTION-2:** Display prisoner's id, name, gender and blocktype of all the prisoners.

**Answer:** SELECT P\_id, P\_name, P\_gender, blocktype FROM Prisoner, Block

WHERE PRISONER.B\_id=Block.B\_id;

☒ Autocommit    Display 10 ▼

```
SELECT P_id, P_name, P_gender, blocktype FROM Prisoner, Block
WHERE PRISONER.B_id=Block.B_id;
```

[Results](#)   [Explain](#)   [Describe](#)   [Saved SQL](#)   [History](#)

P_ID	P_NAME	P_GENDER	BLOCKTYPE
2	Albert L. Bates	Male	Block-A(Male)
1	Jim Jones	Male	Block-A(Male)
3	Charlotte Corday	Female	Block-B(Female)
4	Samantha Lewthwaite	Female	Block-D(Female)
5	Arthur Barker	Male	Block-E(Male)

5 rows returned in 0.01 seconds      [CSV Export](#)

## **View:**

**Question-1:** Create one view called StaffPhone based on the s\_name and s\_phone from the Staff table.

**Answer:** CREATE VIEW StaffPhone as(SELECT S\_name, S\_phone FROM Staff);

☒ Autocommit   Display 10 ▼

CREATE VIEW StaffPhone as(SELECT S\_name, S\_phone FROM Staff);

**Results**   Explain   Describe   Saved SQL   History

View created.

0.00 seconds

**Question-2:** Display all data from StaffPhone.

**Answer:** SELECT \* FROM StaffPhone;

☒ Autocommit   Display 10 ▼

SELECT \* FROM StaffPhone;

**Results**   Explain   Describe   Saved SQL   History

S_NAME	S_PHONE
Zakaria Brooks	1714445555
Joshua Nicole	2223331111
Sasha Grey	4455552233
Jon Rogers	6666887799
Alex Cox	9999666777

5 rows returned in 0.00 seconds   [CSV Export](#)

# Relational Algebra

**Question-1:** Find the name of the Staff whose Id is 1104.

**Answer:**  $\Pi_{s\_name} (\sigma_{s\_id=1104}(\text{Staff}))$

**Question-2:** Find the Staff whose status is Inactive.

**Answer:**  $\Pi_{s\_name} (\sigma_{s\_status=\text{"INACTIVE"}}(\text{Staff}))$

**Question-3:** Find the name of the Staffs who have salary less than 2000.

**Answer:**  $\Pi_{s\_name} (\sigma_{s\_salary < 2000}(\text{Staff}))$

**Question-4:** Find the name of all prisoner.

**Answer:**  $\Pi_{p\_name} (\text{Prisoner})$

**Question-5:** Find the relation of visitor Jack Quaid with Prisoner.

**Answer:**  $\Pi_{v\_relation} (\sigma_{v\_name=\text{"Jack Quaid"}}(\text{Visitor}))$



## **Conclusion**

After a lot of hard work together, we finally created a Prison Management System. Considering current circumstances it was really tough for us to communicate with each other but finally we finished it. Hopefully in future we will be able to create a real Database Management System.