

## Milestone 2

March 3, 2025

Project Group Number on Canvas: ‘Group 41’

<b>Name</b>	<b>Student ID</b>	<b>CS Alias</b>	<b>Preferred Email Address</b>
Vincent Luong	73547515	v8c0o	vincentluong1@hotmail.com
Ahmed Khan	31684178	h6v1y	ahmeddx400@gmail.com
Zain Ali	94391034	k9y0h	szainali284@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

**CPSC 304 Introduction to Relational Databases**  
The University of British Columbia

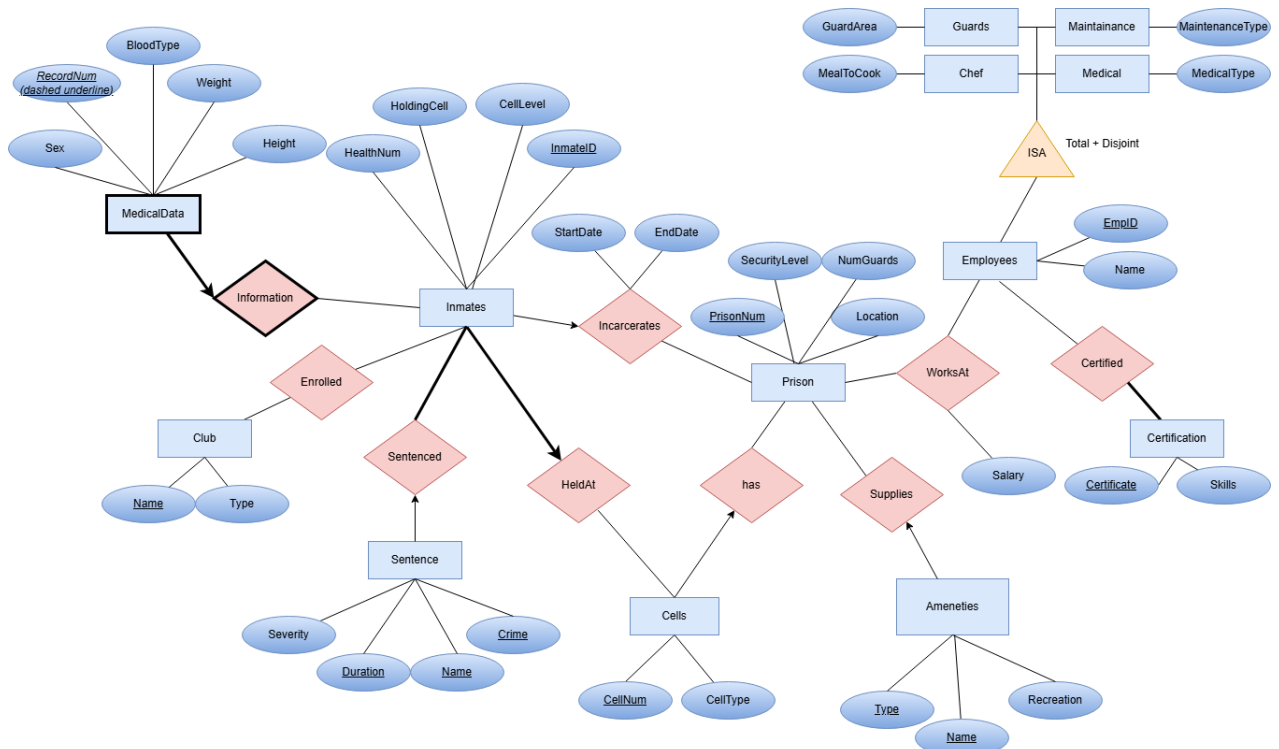
# 1 Project Summary

*A brief summary about our project (2-3 sentences)*

We are developing a prison database management system from the ground up. This system will store and manage essential information about a prison and its inmates while capturing and modeling the facility's internal logistics.

# 2 ER Diagram

*The ER diagram*



We updated RecordNum to be dashed-underlined, introduced additional attributes for Prison and Inmates, and added a Certified relation linked to the Certification object table with three new attributes. Certification now has total participation in Certified. Additionally, we renamed all attributes to follow CamelCase naming conventions.

### 3 Relational Schema

*The translation of the ER diagram to the relational model*

In our following schemas, underlined attributes represents primary keys (PK). while **bolded attributes** indicate foreign keys (FK), and NOT NULL indicating that there must be a value.

- MedicalData(RecordNum: int, **InmateID**: int, BloodType: char, Weight: int, Sex : char, Height: int)
- Inmates(**InmateID**: int NOT NULL, **CellNum**: int, **PrisonNum**: char, HoldingCell: char, CellLevel: int, HealthNum: int, StartDate: date, EndDate: date)
- Club(Name: char, type: char)
- Sentence(Duration: double, Name: char, Crime: char, Severity: int, **InmateID**: int)
- Prison(PrisonNum: int, SecurityLevel: int, NumGuards: int, Location: char)
- Cells(CellNum: int, CellType: char, **PrisonNum**: int)
- Amenities(Type: char, Name: name, Recreation: char, **PrisonNum**: int)
- WorksAt(**PrisonNum**: int, EmpID: int, Salary: int)
- Employees(EmpID: int NOT NULL, Name: char)
- Certification(Certificate: char, skills: char. **EmpID**: int)
- Guards(**EmpID**: int, GuardArea: char)
- Chef(**EmpID**: int, MealToCook: char)
- Maintenance(**EmpID**: int, MaintenanceType: char)
- Medical(**EmpID**: int, MedicalType: char)

## 4 Functional Dependencies

*Identify the functional dependencies in your relations*

1. MedicalData
  - (a)  $\text{RecordNum} \rightarrow \text{BloodType}, \text{Height}, \text{Weight}, \text{Sex}, \text{InmateID}$
2. Inmates
  - (a)  $\text{InmateID} \rightarrow \text{HealthNum}, \text{HoldingCell}, \text{CellLevel}, \text{ClubName}, \text{Duration}, \text{SentenceName}, \text{CrimeName}, \text{CellNum}, \text{PrisonNum}$
  - (b)  $\text{HoldingCell} \rightarrow \text{CellLevel}$
3. Club
  - (a)  $\text{Name} \rightarrow \text{Type}$
4. Sentence
  - (a)  $\text{Duration}, \text{Name}, \text{Crime} \rightarrow \text{severity}, \text{InmateID}$
5. Cells
  - (a)  $\text{CellNum} \rightarrow \text{CellType}, \text{PrisonNum}$
6. Prison
  - (a)  $\text{PrisonNum} \rightarrow \text{SecurityLevel}, \text{NumGuards}, \text{Location}$
  - (b)  $\text{SecurityLevel} \rightarrow \text{NumGuards}, \text{Location}$
7. Amenities
  - (a)  $\text{Type}, \text{Name} \rightarrow \text{Recreation}, \text{PrisonNum}$
8. Certification
  - (a)  $\text{Certificate} \rightarrow \text{Skills}$
9. Guards
  - (a)  $\text{EmpID} \rightarrow \text{GuardArea}$
10. Chef
  - (a)  $\text{EmpID} \rightarrow \text{MealToCook}$
11. Maintenance
  - (a)  $\text{EmpID} \rightarrow \text{MaintenanceType}$
12. Medical
  - (a)  $\text{EmpID} \rightarrow \text{MedicalType}$
13. Incarcerates
  - (a) No non-trivial FD's
14. WorksAt
  - (a) no non-trivial FD's
15. Employees
  - (a)  $\text{Certificate} \rightarrow \text{Skills}$

Writing out our functional dependencies, we can see that the **Inmates** and **Prison** relations include non-trivial primary keys dependencies, and do not Satisfy BCNF; we will decompose them in the next step.

## 5 Normalization

We use the lossless-join BCNF decomposition to decompose our relations to ensure our relations are in BCNF. Since all of our relations are in BCNF except for **Inmates** and **Prison**, we will apply the lossless-join BCNF decomposition algorithm on the following relations:

We first decompose on **Inmates**

1. Inmates(InmateID, ClubName, Duration, SentenceName, CrimeName, CellNum, PrisonNum, HoldingCell, CellLevel, HealthNum)
2. InmateCell(HoldingCell, CellLevel)
3. InmatesInfo(InmateID, ClubName, Duration, SentenceName, CrimeName, CellNum, PrisonNum, HoldingCell, HealthNum)

(a) by  $\text{HoldingCell} \rightarrow \text{CellLevel}$

Next, we decompose on **Prison**

1. Prison(PrisonNum, SecurityLevel, NumGuards, Location)
2. PrisonSecurity(SecurityLevel, NumGuards, Location)
3. PrisonInfo(PrisonNum, SecurityLevel)

(a) by  $\text{SecurityLevel} \rightarrow \text{NumGuards}, \text{Location}$

Combining our decomposed relations together with our original relational schemas, we get:

- MedicalData(RecordNum: int, **InmateID**: int, BloodType char, Weight: int, Sex : char, Height: int)
- InmatesInfo(InmateID: int NOT NULL, **CellNum**: int, **PrisonNum**: char, **HoldingCell**: char, HealthNum: int)
- InmatesCell(HoldingCell: char, CellLevel: int)
- Club(Name: char, type: char)
- Sentence(Duration: double, Name: char, Crime: char, Severity: int, **InmateID**: int)
- Incarcerates(**InmateID**: int, **PrisonNum**: int, StartDate: date, EndDate: date)
- PrisonInfo(PrisonNum: int, **SecurityLevel**: int)
- PrisonSecurity(SecurityLevel: int, NumGuards: int, Location: char)
- Cells(CellNum: int, CellType: char, **PrisonNum**: int)
- Amenities(Type: char, Name: name, Recreation: char, **PrisonNum**: int)
- WorksAt(**PrisonNum**: int, EmpID: int, Salary: int)
- Employees(EmpID: int NOT NULL, Name: char)
- Certification(Certificate: char, skills: char, **EmpID** : int)
- Guards(EmpID: int, GuardArea: char)
- Chef(EmpID: int, MealToCook: char)
- Maintenance(EmpID: int, MaintenanceType: char)
- Medical(EmpID: int, MedicalType: char)

## 6 SQL DDL

Below are the SQL DDL tables made from the Normalized relation:

```
CREATE TABLE Inmates1(  
    InmateID int NOT NULL,  
    HoldingCell varchar(50),  
    HealthNum int,  
    PRIMARY KEY(InmateID),  
    FOREIGN KEY(Name) REFERENCES Club ON DELETE CASCADE,  
    FOREIGN KEY(Duration, Name, Crime) REFERENCES SENTENCE ON DELETE CASCADE,  
    FOREIGN KEY(CellNum) REFERENCES Cells ON DELETE CASCADE,  
    FOREIGN KEY(PrisonNum), REFERENCES Prison1 ON DELETE CASCADE  
);
```

```
CREATE TABLE MedicalData(  
    RecordNum int,  
    BloodType char(255),  
    Weight int,  
    Sex varchar(255),  
    Height int,  
    PRIMARY KEY(RecordNum, InmateID)  
    FOREIGN KEY (InmateID) REFERENCES Inmates1 ON DELETE CASCADE  
);
```

```
CREATE TABLE Inmates2(  
    HoldingCell varchar(50) PRIMARY KEY,  
    CellLevel int  
);
```

```
CREATE TABLE Club(  
    Name varchar(255) PRIMARY KEY,  
    type varchar(255)  
);
```

```
CREATE TABLE Sentence(  
    Duration double,  
    Name char(50),  
    Crime char(255),  
    Severity int,  
    PRIMARY KEY(Duration, Name, Crime),  
    FOREIGN KEY(InmateID) REFERENCES Inmates1 ON DELETE CASCADE  
);
```

```

CREATE TABLE Incarcerates(
    StartDate date,
    EndDate date,
    PRIMARY KEY(InmateID, PrisonNum),
    FOREIGN KEY(InmateID) REFERENCES Inmates1 ON DELETE CASCADE,
    FOREIGN KEY(PrisonNum) REFERENCES Prison1 ON DELETE CASCADE
);

CREATE TABLE Prison1(
    PrisonNum int PRIMARY KEY,
    SecurityLevel int
);

CREATE TABLE Prison2(
    SecurityLevel int PRIMARY KEY,
    NumGuards int,
    Location varchar(255)
);

CREATE TABLE Cells(
    CellNum int PRIMARY KEY,
    CellType varchar(255),
    FOREIGN KEY(PrisonNum) REFERENCES Prison1
);

CREATE Table Amenities(
    Type varchar(255),
    Name varchar(255),
    Recreation varchar(255),
    PRIMARY KEY(Type, Name),
    FOREIGN KEY(PrisonNum) REFERENCES Prison1 ON DELETE CASCADE
);

CREATE TABLE WorksAt(
    Salary int,
    PRIMARY KEY(PrisonNum, EmpID),
    FOREIGN KEY(PrisonNum) REFERENCES Prison1 ON DELETE CASCADE,
    FOREIGN KEY(EmpID) REFERENCES Employees ON DELETE CASCADE
);

```

```

CREATE TABLE EMPLOYEES(
    EmpID int PRIMARY KEY NOT NULL,
    name varchar(255)
);

CREATE TABLE Certification(
    Certificate varchar(255) PRIMARY KEY,
    Skills varchar(255)
);

CREATE TABLE Guards(
    GuardArea varchar(255),
    PRIMARY KEY(EmpId),
    FOREIGN KEY(EmpID) REFERENCES Employees ON DELETE CASCADE
);

CREATE TABLE Chef(
    MealToCook varchar(255),
    PRIMARY KEY(EmpId),
    FOREIGN KEY(EmpID) REFERENCES Employees ON DELETE CASCADE
);

CREATE TABLE Maintenance(
    MaintenanceType varchar(255),
    PRIMARY KEY(EmpId),
    FOREIGN KEY(EmpID) REFERENCES Employees ON DELETE CASCADE
);

CREATE TABLE Medical(
    MedicalType varchar(255),
    PRIMARY KEY(EmpId),
    FOREIGN KEY(EmpID) REFERENCES Employees ON DELETE CASCADE
);

```



## 7 SQL DDL Insertion

```
--MedicalData(RecordNum, BloodType, Weight, Sex, Height)
INSERT INTO MedicalData VALUES(1000001, "O-", 195, "M", 189);
INSERT INTO MedicalData VALUES(1000231, "O+", 230, "M", 170);
INSERT INTO MedicalData VALUES(1000151, "B-", 150, "F", 156);
INSERT INTO MedicalData VALUES(1000291, "AB+", 220, "F", 170);
INSERT INTO MedicalData VALUES(1000901, "A+", 160, "M", 165);

--Inmates1(InmateID, HoldingCell, HealthNum)
INSERT INTO Inmates1 VALUES(20000001, "REGULAR", 10005562);
INSERT INTO Inmates1 VALUES(20000021, "DEATH ROW", 16702222);
INSERT INTO Inmates1 VALUES(20000626, "DEATHROW", 19023567);
INSERT INTO Inmates1 VALUES(20005502, "SOLITARY", 15420213);
INSERT INTO Inmates1 VALUES(20001237, "REGULAR", 10111111);

--Inmates2(HoldingCell, CellLevel)
--HoldingCell --> CellLevel
INSERT INTO Inmates2 VALUES("REGULAR", 2);
INSERT INTO Inmates2 VALUES("REGULAR", 2);
INSERT INTO Inmates2 VALUES("DEATH ROW", 1);
INSERT INTO Inmates2 VALUES("SOLITARY", 3);
INSERT INTO Inmates2 VALUES("REGULAR", 2);

--Club(Name, Type)
INSERT INTO Club("Weight-Lifting", "Physical");
INSERT INTO Club("Dance", "Physical");
INSERT INTO Club("Bird-Watching", "Enjoyment");
INSERT INTO Club("Book", "Education");
INSERT INTO Club("Running", "Physical");

--Sentence(Duration, Name, Crime, Severity)
INSERT INTO Sentence(2.5, "Assault with a deadly weapon", "Assault", 7);
INSERT INTO Sentence(0.5, "Tax Fraud", "Tax Invasion", 1);
INSERT INTO Sentence(25, "Double Homicide", "Homicide", 9);
INSERT INTO Sentence(3, "Vehicle Manslaughter", "Homicide", 5);
INSERT INTO Sentence(5, "Breaking-and-Entering", "Assault", 6);

--Incarcerates(StartDate, EndDate)
INSERT INTO Sentence("2008/02/22", "2009/02/22");
INSERT INTO Sentence("2002/01/22", "2012/01/22");
INSERT INTO Sentence("2025/02/11", "2075/02/11");
INSERT INTO Sentence("1975/07/14", "2025/07/14");
INSERT INTO Sentence("2010/09/22", "2015/09/22");
```

```

--Prison1(PrisonNum, SecurityLevel)
INSERT INTO Prison1(132, 5);
INSERT INTO Prison1(101, 2);
INSERT INTO Prison1(106, 7);
INSERT INTO Prison1(192, 10);
INSERT INTO Prison1(123, 1);

--Prison2(SecurityLevel, NumGuards, Location)
INSERT INTO Prison2(5, 10000, "Vancouver, BC");
INSERT INTO Prison2(10, 50000, "Washington, DC");
INSERT INTO Prison2(1, 1000, "Toronto, ON");
INSERT INTO Prison2(3, 1750, "Surrey, BC");
INSERT INTO Prison2(7, 22000, "Los Angeles, CA");

--Cells(CellNum, CellType)
INSERT INTO Cells(1002, "DEATH ROW");
INSERT INTO Cells(2002, "REGULAR");
INSERT INTO Cells(3002, "SOLITARY");
INSERT INTO Cells(2025, "REGULAR");
INSERT INTO Cells(2102, "REGULAR");

--Amenities(Type, Name, Recreation)
INSERT INTO Amenities("Physical", "Gym", "General Rec");
INSERT INTO Amenities("Physical", "Pool", "General Rec");
INSERT INTO Amenities("Education", "Library", "Learning");
INSERT INTO Amenities("Education", "Computer Room", "Electronics");
INSERT INTO Amenities("Entertainment", "Cafeteria", "Food and Entertainment");

--WorksAt(Salary)
INSERT INTO WorksAt(100000);
INSERT INTO WorksAt(70000);
INSERT INTO WorksAt(50000);
INSERT INTO WorksAt(40000);
INSERT INTO WorksAt(60000);

--Employees(EmpID, Name)
INSERT INTO Employees(10005, "Cell Guard");
INSERT INTO Employees(10006, "Solitary Confinement Guard");
INSERT INTO Employees(20002, "Sous Chef");
INSERT INTO Employees(30002, "Janitor");
INSERT INTO Employees(40023, "Nurse");

```

```

--Certification(Certificate, Skills)
INSERT INTO Certification("Food Safety", "Culinary");
INSERT INTO Certification("Legal Studies", "Policing");
INSERT INTO Certification("Plumbing", "Janitorial");
INSERT INTO Certification("Police Academy", "Policing");
INSERT INTO Certification("Bachelors of Nursing", "Nursing");

--Guards(GuardArea)
INSERT INTO Guards("North Sector");
INSERT INTO Guards("South Sector");
INSERT INTO Guards("Solitary");
INSERT INTO Guards("West Sector");
INSERT INTO Guards("East Sector");

--Chef(MealToCook)
INSERT INTO Chef("Mac N Cheese");
INSERT INTO Chef("Pizza");
INSERT INTO Chef("Chicken Breast");
INSERT INTO Chef("Chicken Noodle Soup");
INSERT INTO Chef("Rice and Curry");

--Maintenance(MaintenanceType)
INSERT INTO Maintenance("Janitor");
INSERT INTO Maintenance("Plumbing");
INSERT INTO Maintenance("Pipes");
INSERT INTO Maintenance("Custodian");
INSERT INTO Maintenance("Checkup");

--Medical(MedicalType)
INSERT INTO Medical("Nurse");
INSERT INTO Medical("Doctor");
INSERT INTO Medical("Patient Care");
INSERT INTO Medical("Pharmacologist");
INSERT INTO Medical("Pharmacy Assistant");

```

## 8 AI Declaration

Yes, AI has been used to spellcheck and retype our words for the following **The ER Diagram Changes**. We gave it the following prompt to ChatGPT:

Rewrite this: We updated RecordNum to be dashed-underlined, introduced additional attributes for Prison and Inmates, and added a Certified relation linked to the Certification object table with three new attributes. Certification now has total participation in Certified. Additionally, we renamed all attributes to follow CamelCase naming conventions.

Everything else, we used lecture material and TA help to confirm our concerns.