King Abdulaziz University
Faculty of Computing and Information Technology
Computer Science Department

CPCS 241

Databases

Group Project – Phase III

Student Names: …….

Student Numbers: ……

Group Number: ……

Semester: Spring 2022

Submission Date: Monday 15/10/1443 H, **16/05/2022** G

Submission Time: DAR & GAR @ 9:30 AM, IAR @ 12:30 PM

Submission Format: Paper copy + Soft copy via BB

Criteria	Max Mark		Obtained Mark
Normalization	1		
Creation of Tables, Relationships and Constraints	1		
Data Population	1		
Queries	1		
Report Organization	1		
Total	5		

Instructors: Dr. Lama Al Khuzayem and Dr. Rsha Mirza

lalkhuzayem@kau.edu.sa , rmirza@kau.edu.sa

[City Jail Management System]

DB Design

CPCS241 – Database I – Spring 2022 – Group Project

Group No: 11

Student Name	Student Number
Fai Badr Alnagrani	2006268
Shrooq Kaabi	2205003
Shahad Hatem Balkhasher	2006676
Majd Saleh	1816903
Wajd Alharbi	2007057

Contents

PART I: ANALYSIS 3
1 Problem Definition and Data Requirements 3
1.1 Problem Description 3
1.2 Data Requirements 4
1.3 Business Rules 7
1.4 Intended Output of the System 10
PART II: DB DESIGN 11
2 EER Diagram Design 11
2.1 EER Diagram 11
2.2 Design of Business Rules 12
3 ER-to-logical schema mapping 16
3.1 Mapping of Regular Entity Types 16
3.2 Mapping of Weak Entity Types 20
3.3 Mapping of Binary 1-1 Relationship Types 22
3.4 Mapping of Binary 1-N Relationship Types 22
3.5 Mapping of Binary M-N Relationship Types 27
3.6 Mapping of Multivalued Attributes 27
3.7 Mapping of N-ary Relationship Types 27
3.8 Schema Diagram 28
4 Normalization 29
4.1 First Normal Form 29
4.2 Second Normal Form 30
4.3 Third Normal Form 33
5 Final DB Schema Diagram 35
PART III: IMPLEMENTATION 36
6 Table Creation Script 36
6.1 < WARDEN > TABLE 36
6.2 < JOB_SALARY > TABLE 37
6.3 < STAFF > TABLE 37
6.4 < FLOOR> TABLE 38
6.5 < PRISON_OFFICERS > TABLE 38
6.6 < PRISON_COOK > TABLE 38
6.7 < HAED_DOCTOR > TABLE 39
6.8 < YARD_OFFICER > TABLE 39
6.9 < CELLS > TABLE 39

6.10 < PRISONERS > TABLE 40
6.11 < HEALTH_RECORDS > TABLE 41
6.12 < CRIMINAL_RECORDS > TABLE 42
6.13 < VISITS > TABLE 43
6.14 < ACTIVITY_REWARD > TABLE 43
6.15 < ACTIVITIES > TABLE 44
6.16 < MANAGED > TABLE 44
6.17 < CELLS_FLOOR > TABLE 45
6.18 < WARDEN_FLOOR > TABLE 45
6.19 < PUNISHMENT > TABLE 46
6.20 < LAW_VIOLATORS> TABLE 46
7 Constraints Script 47
8 Queries and Transactions 48
8.1 <Average salaries > 48
8.2 *<Specify prisoners with certain characteristics>* 49
8.3 *<Specify robbers>* 50
8.4 *<Salary Raise>* 51
8.5 *<Specify jobs with less than two members >* 52
8.6*<Total Number of Staff Members Per Job Type>* 53
8.7*<The Largest Number of Staff Members in a Job Type>* 54
8.8 Update Example 55
8.8.1 example 55
8.8.2 example 56
8.9 Delete Example 57
8.9.1 example 57
8.9.2 example 58
9 APPENDIX 59
WARDEN TABLE 60
JOB_SALARY TABLE 60
STAFF TABLE 60
PRISON_COOK TABLE 61
YARD_OFFICER TABLE 62
CELLS TABLE 62
PRISONERS TABLE 64
HEALTH_RECORDS TABLE 64
CRIMINAL_RECORDS TABLE 65
VISITS TABLE 66
ACTIVITY_REWARD TABLE 66

ACTIVITIES TABLE 66
MANAGED TABLE 67
CELLS_FLOOR TABLE 68
WARDEN_FLOOR TABLE 68
PUNISHMENT TABLE 68
LAW_VIOLATORS TABLE 69

PART I: Analysis

1 Problem Definition and Data Requirements

1.1 Problem Description

As time has passed, the technological revolution has been used in the justice system to assist it. The utilization of a proper management system will without doubt be effective in improving data manipulation. To manage the city jail system, we designed a database to store, maintain, and access all the information about the prisoners detained inside the prison, which will enhance the way the prison is administrated, and help cities efficiently track and process inmates from the day they arrive to the day they are released. Every jail system consists of police officers (people in control) and prisoners (people under authority). In this system, prisoners are being watched and managed by police officers, creating a strict atmosphere that prevents prisoners from escaping or committing chaos.

1.2 Data Requirements

1. CELLS

- Each CELL has a unique cell number.
- Each CELL has a capacity (the maximum number of prisoners that can occupy a cell).
- Each CELL has a state (cell can be either occupied or unoccupied).

2. PRISONERS

- Each PRISONER has a prisoner number (a number that uniquely identifies the prisoner inside the prison).
- Each PRISONER has a composite attribute name (first name, last name).
- Each PRISONER has a date of arrival (the date on which the prisoner has arrived at the prison).
- Each PRISONER has a date of release (the date on which the prisoner is supposed to leave the prison).

3. HEALTH RECORDS

- HEALTH RECORDS is a weak entity of PRISONERS.
- Each HEALTH RECORD has the prisoner's name which is the partial key.
- Each HEALTH RECORD has the prisoner's height.
- Each HEALTH RECORD has the prisoner's weight.
- Each HEALTH RECORD has the prisoner's date of birth.
- Each HEALTH RECORD has the prisoner's state (single, married, widowed, etc.).
- Each HEALTH RECORD has the prisoner's blood type.
- Each HEALTH RECORD has the prisoner's disease history (a record of a prisoner's symptoms, illnesses, conditions, and alterations in development).

4. STAFF

- Each STAFF member has a staff ID.
- Each STAFF member has a composite attribute name (first name, last name).

- Each STAFF member has gender.
- Each STAFF member has a job type.
- Each STAFF member has a salary.
- Each STAFF member has an address.
- STAFF has specializations: {PRISON OFFICERS, YARD OFFICERS, HEAD DOCTOR, and PRISON COOK} based on the STAFF member's job type.

5. VISITS

- Each VISIT has the visitor's ID.
- Each VISIT has the visitor's name.
- Each VISIT has the visitor's address.
- Each VISIT has the visitor's phone number.
- Each VISIT has anumberof visits (Each prisoner has a certain number of visits).
- Each VISIT has the time of the visit (the time at which the visit happens).
- Each VISIT has the duration of the visit (the time during which the visit continues).

6. FLOOR

- Each FLOOR has a floor number.
- Each FLOOR has a floor's type (either cell's floor or warden's floor).

7. CELL'S FLOOR

- CELL'S FLOOR IS-A FLOOR that has the number of cells as an attribute.

8. WARDEN'S FLOOR

- WARDEN'S FLOOR IS-A FLOOR that has WARDEN'S office number as an attribute.

9. CRIMINAL RECORDS

- A CRIMINAL RECORD is a weak entity.
- Each CRIMINAL RECORD has the prisoner's name as a partial key.

- Each CRIMINAL RECORD has the prisoner 's crimes.

10. ACTIVITIES

- PRISONERS can be assigned an ACTIVITY (a helpful activity).
- Each **ACTIVITY** has an activity number.
- Each **ACTIVITY** has an activity type (cleaning, laundry, community work, etc.).
- Each **ACTIVITY** has a reward (payment, early discharge).

11. PUNISHMENT

- It is a result of violating the law or committing any crime inside the prison.
- Each PUNISHMENT has a Punishment number.
- Each PUNISHMENT has a punishment type (withdrawing certain privileges, such as seeing visitors, forced labor, etc.).

12. LAW VIOLATORS

- LAW VIOLATERS is a weak entity.
- Each LAW VIOLATOR has the name of the prisoner as a partial key.

13. WARDEN

- The PRISON WARDEN is an official in charge of the prison including STAFF and PRISONERS.
- The WARDEN has a warden ID.
- The WARDEN has a composite attribute name (first name, last name).
- The WARDEN has a salary.
- The WARDEN has a gender.
- The WARDEN has an address.
- The WARDEN has a password to the system.

14. CANTEEN

15. PRISON YARD

16. HEALTHCARE

17. PRISON COOK

18. HEAD DOCTOR

19. YARD OFFICERS

20. PRISON OFFICERS

1.3 Business Rules

CELLS:

- Each prisoner must be [PLACED] in one CELL, each CELL may have many PRISONERS [PLACED] in it (meaning: all prisoners must be placed in cells but a cell can have one or more prisoners).
- A cell is considered fully occupied only if it is being occupied by two PRISONERS otherwise, a new PRISONER will be [PLACED] to the cell by the system.

PRISONERS:

- Each prisoner must be [PLACED] in one CELL (must be within each cell's capacity).
- The constraints are mentioned in CELLS rules (above).

HEALTH RECORDS:

- Each PRISONER must [**HAS]** one HEALTH RECORD, each HEALTH RECORD must belong to one PRISONER.
- Prisoners' HEALTH RECORDS are related to their existence (when a prisoner is released, their health record is automatically deleted).

STAFF:

- Each PRISONER must be [**MANAGED]** by many STAFF members, each STAFF member may manage many PRISONERS.
- Each PRISON COOK must [**WORKS_IN]** one CANTEEN, each CANTEEN must have one PRISON COOK.

- Each YARD OFFICER must [**OBSERVES]** one PRISON YARD, each PRISON YARD must be observed by many YARD OFFICERS.

- Each HEAD DOCTOR must [**HEADS]** one HEALTHCARE, each HEALTHCARE must have one HEAD DOCTOR.

- HEAD DOCTOR IS-A STAFF

- PRISON OFFICERS IS-A STAFF

- PRISON COOK IS-A STAFF

- YARD OFFICERS IS-A STAFF

VISITS:

- Each PRISONER may **[RECEIVE]** many VISITS, each VISIT must be for one PRISONER.

FLOOR:

- Each FLOOR must be **[SURVEILLANCED]** by many PRISON OFFICERS, each PRISON OFFICER must **[SURVEILLANCED]** one CELLS' FLOOR.
- FLOORS are further grouped into {CELL'S FLOOR, WARDEN'S FLOOR} based on the floor's type.

CELL'S FLOOR:

- Each CELL'S FLOOR must [**CONTAINS]** many CELLS, each CELL must be on one CELL'S FLOOR.

WARDEN'S FLOOR:

- This entity has no relationships.

CRIMINAL RECORDS:

- Each PRISONER must **[OWNS]** one CRIMINAL RECORDS, each CRIMINAL RECORD must belong to one PRISONER.

ACTIVITIES:

- Each ACTIVITY must be [**ASSIGNED_TO**] one PRISONER, each PRISONER may have manyACTIVITIES.

PUNISHMENT:

- Each PUNISHMENT must [**DISCIPLINES]** one LAW VIOLATOR, each LAW VIOLATOR must be disciplined by one punishment.

WARDEN:

- Each WARDEN must [**OVERSEES]** many STAFF members, each STAFF member must be overseen by one WARDEN.
- Each WARDEN must [**DECIDES]** many PUNISHMENTS, Each PUNISHMENT must be decided by one warden.

1.4 Intended Output of the system

- **Transaction**

– Insert staff members into the system.
– Delete a prisoner from the system.
– Delete a staff member from the system.
– Assign a prisoner to an available cell
– Update cell state
– Assign a prison officer on a specific floor
– Update prisoners' health record
– Update health recorders for prisoners
– Insert a crime in a prisoner's criminal record
– Assign a prisoner as a law violator
– Insert a new visitor's details
– Update a yard officer
– Assign an activity for a prisoner
– Update number of visits
– Assign a new prison cook for the canteen
–etc.

- **Queries/output**

– Search for a cell number by a cell type.
– Display visitors of a specific prisoner.
– Display prisoners' activity type.
– Search criminal records for a prisoner by the prisoner ID.
– Display occupied cells.
– Display number of unoccupied cells.
– Count the remaining years until the release date for a prisoner.
– Display the remaining time until visiting hours are over.
– Count number of staff-by-staff ID.
– Display reward for a prisoner by activity number.
– Display all prisoners with their criminal records.
– Search address by staff ID.
– Retrieve the average salary of each job type with an average salary greater than the specified department average salary.
– Display the Prisoner number of all prisoners who will be released after the date (1-1-2026), and their initials are not the letter (R).
– Display ID, first name, and last name of prisoners who have committed robbery.
– Raise the salary of all employees by 10%.
– Display jobs that have less than 2 members working in them including their names.
– ….....etc.

PART II: DP DESIGN

2 EER Diagram design

2.1 EER Diagram

- **Total participation is shown in blue**

2.2 Design of Business Rules

Business Rules	Design Decisions	Justifications (if any)
Many PRISONERS must be **[PLACED]** in one CELL.	N:1 binary relationship between PRISONERS and CELL.	Each PRISONER must be **[PLACED]** in one CELL, each CELL may have many PRISONERS **[PLACED]** in it. It is total participation on the PRISONERS' side since all PRISONERS will be placed in CELLS, and it is partial participation on the CELLS' side since CELLS can be occupied by one or more PRISONERS.
Each PRISONER must [**HAS]** one HEALTH RECORD that is related to them as a weak entity.	1:1 binary identifying relationship between PRISONER and HEALTH RECORDS.	Each PRISONER must **[HAS]** one HEALTH RECORD, each HEALTH RECORD must belong to one PRISONER If a PRISONER does not exist, their HEALTH RECORD is unnecessary, since its existence depends on a PRISONER, it is shown in total participation on both sides.
STAFF members must have a job type.	Superclass/subclass relationship with disjointness and completeness constraints.	Total participation since each STAFF member must have a job type. Each STAFF member must have one job type; therefore, it is disjointed**.**
many PRISONERS must be [**MANAGED]** by many prison STAFF.	N: M binary relationship between PRISONERS and STAFF	Each PRISONER must be **[MANAGED]** by many STAFF members, each STAFF member may manage many PRISONERS. It is total participation on the PRISONERS' side because PRISONERS will be managed only by STAFF members. On the other hand, it is partial participation since STAFF has different jobs.

Each PRISON COOK must [**WORKS_IN]** in one CANTEEN.	1:1 binary relationship between PRISON COOK and CANTEEN.	Each PRISON COOK must **[WORKS_IN]** one CANTEEN, each CANTEEN must have one PRISON COOK It is total participation on both sides since PRISON COOK must work only in the CANTEEN and the CANTEEN must have only one PRISON COOK.
Many YARD OFFICERS [**OBSERVE]** one PRISON YARD.	N:1 binary relationship between YARD OFFICERS and PRISON YARD.	Each YARD OFFICER must **[OBSERVES]** one PRISON YARD, each PRISON YARD must be observed by many YARD OFFICERS. Many YARD OFFICERS must observe one PRISON YARD, and the only PRISON YARD must be observed by many YARD OFFICERS, hence, it is total participation on both sides.
One HEAD DOCTOR must **[HEADS]** the prisons HEALTHCARE	1:1 binary relationship between HEAD DOCTORS and HEALTHCARE	Each HEAD DOCTOR must **[HEADS]** one HEALTHCARE, each HEALTHCARE. It is total participation on both sides since the only HEAD DOCTOR will head the only healthcare, and the healthcare will only have one head doctor.
Each PRISONER can **[RECEIVE]** a certain number of VISITS a month.	1: N binary relationship between PRISONER and VISIT.	Each PRISONER may **[RECEIVE]** many VISITS, each VISIT must be for one PRISONER. It is partial participation on the PRISONERS' side because not all PRISONERS will receive VISITS and it is total participation on the VISITS' side because many VISITS can be received by one PRISONER.

FLOOR must have a floor type.	Superclass/subclass relationship with disjointness and completeness constraints.	Total participation since every FLOOR has a floor type. Each FLOOR has one floor type; therefore, it is disjointed.
Each FLOOR must be [**SURVEILLANCED**] by many PRISON OFFICERS.	1: N binary relationship between FLOOR and PRISON OFFICERS.	Each FLOOR must be [**SURVEILLANCED**] by many PRISON OFFICERS, each PRISON OFFICER must [**SURVEILLANCED**] one CELLS' FLOOR. All FLOORS will be surveillanced by all PRISON OFFICERS and all PRISON OFFICERS will surveillance all FLOORS. Hence, it is total participation on both sides.
Each CELL'S FLOOR must [**CONTAINS**] many CELLS.	1: N binary relationship between CELL'S FLOOR and CELLS.	Each CELL'S FLOOR must [**CONTAINS**] many CELLS, each CELL must be on one CELL'S FLOOR. It is total participation on both sides because all CELL'S FLOORS will contain CELLS and all CELLS will be on CELL'S FLOOR.
Each PRISONER must [**OWNS**] one CRIMINAL RECORDS.	1:1 binary relationship between PRISONERS and CRIMINAL RECORDS.	Each PRISONER must **[OWNS]** one CRIMINAL RECORD and each CRIMINAL RECORD belongs to only one PRISONER. It is total participation on both sides because all prisoners must own a CRIMINAL RECORD and all CRIMINAL RECORDS belong to PRISONERS.

Many ACTIVITIES may be [**ASSIGNED_TO]** each PRISONER.	N: 1 relationship between ACTIVITIES and PRISONERS.	Each ACTIVITY must be [**ASSIGNED_TO**] one PRISONER, each PRISONER may have many ACTIVITIES. It is total participation in the ACTIVITY side because all activities can only be assigned to PRISONERS. On the PRISONERS' side, it is partial participation since not all PRISONERS will be assigned ACTIVITIES.
One PUNISHMENT must [**DISCIPLINES]** each LAW VIOLATOR LAW VIOLATOR IS related to PUNISHMENT as a weak entity.	1:1 binary identifying relationship between PUNISHMENT and LAW VIOLATERS.	Each PUNISHMENT must [**DISCIPLINES**] one LAW VIOLATOR, each LAW VIOLATOR must be disciplined by one punishment. If a LAW VIOLATOR does not exist, the PUNISHMENT is unnecessary, because its existence depends on the LAW VIOLATOR, it is shown in total participation.
One prison WARDEN must [**OVERSEES**] all STAFF members.	1: N binary relationship between WARDEN and STAFF.	Each WARDEN must [**OVERSEES**] many STAFF members, each STAFF member must be overseen by one WARDEN. Total participation on both sides because the only WARDEN will oversee all STAFF members and all STAFF members will be overseen by the only WARDEN
WARDEN must [**DECIDES**] all PUNISHMENTS.	N:1 binary relationship between PUNISHMENT and WARDEN.	Each WARDEN must **[DECIDES]** many PUNISHMENTS, Each PUNISHMENT must be decided by one warden. Total participation on both sides because the WARDEN will decide all the PUNISHMENTS, and all PUNISHMENTS will only be decided by the WARDEN.

3 ER-to-logical schema mapping

3.1 Mapping of Regular Entity Types

- For each regular entity create a new relation and include all the simple attributes, then choose one of the key attributes as the primary key.

CELLS

Cell's number	Cell's capacity	Cell's state

PRISONERS

Prisoner number	First name	Last name	Date of arrival	Date of release

- We used the general way to map the STAFF specialization (EER) since it will benefit us in the implementation phase.

STAFF

Staff member's ID	**First name**	**Last name**	**Gender**	**Job type**	**Salary**	**Address**

PRISON COOK

ID

HEAD DOCTOR

ID

YARD OFFICERS

ID

PRISON OFFICERS

ID

VISITS

Visitor ID	Visitor Name	Visitor Phone number	Visitor Address	Duration Of visit	Time of visits	Number of visits

FLOOR

Floor number	Floor's type

CELL'S FLOOR

F_number	Number of cells

WARDEN'S FLOOR

F_number	office number

ACTIVITIES

Activity number	Reward	Activity type

PUNISHMENT

Punishment number	Punishment type

WARDEN

Warden ID	First name	Last name	Gender	Address	Password	Salary

3.2 Mapping of Weak Entity Types

- Include in the weak entity the primary key of the identifying entity as a
  The foreign key and the primary key of the weak entity are the combinations
  of the primary key of the identifying entity and the partial key of the weak entity.

PRISONER

Prisoner number	First name	Last name	Date of arrival	Date of release

HEALTH RECORDS

Prisoner's name	P_number	Blood type	State	Date of birth	Weight	Height	Disease history

PRISONERS

Prisoner number	First name	Last name	Date of arrival	Date of release

CRIMINAL RECORDS

Prisoner's name	P_number	Prisoner's crime

PUNISHMENT

Punishment number	Punishment type

LAW VIOLATORS

Prisoner's name	Punish_number

3.3 MAPPING OF BINARY 1:1 RELATIONSHIP TYPES

• Shown in (Mapping of Weak Entity Types and Regular Entity Types).

3.4 MAPPING OF 1:N RELATIONSHIP TYPES

- To map the binary 1:N relationship, we include a foreign key on the N side.

CELLS

Cell number	**Cells state**	**Cell capacity**

PRISONERS

Prisoner number	**C_number**	**First name**	**Last name**	**Date of arrival**	**Date of release**

PRISONERS

Prisoner number	First name	Last name	Date of arrival	Date of release

VISITS

Visitor ID	P_number	Visitor Address	Phone Number	Visitor Name	Duration Of visit	Time of visit	Number of visits

Cell's floor

Floor number	Number of Cells

Cells

Cell number	F_number	Cells state	Cell capacity

FLOOR

Floor number	Floor's type

PRISON OFFICERS

ID	F_number

PRISONERS

Prisoner number	First name	Last name	Date of arrival	Date of release

ACTIVITIES

Activity number	P_number	Reward	Activity type

WARDEN

Warden ID	First name	Last name	Gender	Address	Password	Salary

STAFF

Staff member's ID	W_ID	First name	Last name	Gender	Job type	Salary	Address

WARDEN

Warden ID	First name	Last name	Gender	Address	Password	Salary

PUNISHMENT

Punishment number	W_ID	Punishment type

3.5 MAPPING OF M:N RELATIONSHIP TYPES

- Create a new relation for the relationship then combine the primary keys of both entities to form the primary key of the new relation.

PRISONERS

Prisoner number	**First name**	**Last name**	**Date of arrival**	**Date of release**

STAFF

Staff ID	**First name**	**Last name**	**Gender**	**Job type**	**Salary**	**Address**

MANAGED

P_number	**S_ID**

3.6 Mapping of multivalued attributes:

• None

3.7 Mapping of n-ary relationship types:

• None

3.8 Schema diagram

4 Normalization

4.1 First Normal Form

In the First normal form (1NF), composite, multivalued attributes, and nested relations are not allowed. Hence, we only include attributes with atomic values in their domains. Our schema has neither nested relation nor multivalued attributes. We only have one composite attribute which is (name) each sub value was given in a column. For that reason, we can conclude that our relations do not violate any guidelines for the first normal form.

STAFF

Staff member's ID	W_ID	First name	Last name	Gender	Job type	Salary	Address

PRISON COOK

ID

HEAD DOCTOR

ID

YARD OFFICERS

ID

PRISON OFFICES

ID	F_number

PRISONERS

Prisoner number	C_number	First name	Last name	Date of arrival	Date of release

HEALTH RECORDS

Prisoner's name	P_number	State	Blood type	Date of birth	Weight	Height	Disease history

CRIMINAL RECORDS

Prisoner's name	P_number	Prisoner's crime

VISITS

Visitor ID	P_number	Visitor Name	Visitor Phone number	Visitor Address	Duration Of visit	Time of visits	Number of visits

ACTIVITIES

Activity number	Reward	Activity type	P_number

MANAGED

P_number	S_ID

FLOOR

Floor number	Floor's type

CELL'S FLOOR

F_number	Number of cells

WARDENS FLOOR

Office number	F_number

CELLS

Cell's number	Cell's capacity	Cell's state	F_number

PUNISHMENT

Punishment number	W_ID	Punishment type

LAW VIOLATORS

Prisoner's name	Punish number

WARDEN

Warden ID	First name	Last name	Gender	Address	Password	Salary

4.2 Second Normal Form

To make our relational schema in Second normal form [2NF], every non-prime attribute should be fully functionally dependent on the primary key. Full FD: means that the removal of the primary key attribute (or an attribute that is part of a PK) results in losing the functional dependency.

If the relation has a single attribute only, then it will be in a Second Normal Form automatically.

STAFF

Staff member's Id	**W_ID**	**First name**	**Last name**	**Gender**	**Job type**	**Salary**	**Address**

{Staff member's ID} -> W_ID, Fname, Lname, gender, job type, salary and address are all functionally dependent [FD] on the Primary key, and is it a full FD.

PRISONERS

Prisoner number	**C_number**	**First name**	**Last name**	**Date of arrival**	**Date of release**

{Prisoner Number} -> C_number, first name, last name, date of arrival and date of release are full FD.

HEALTH RECORDS

Prisoner's name	**P_number**	**State**	**Blood type**	**Date of birth**	**Weight**	**Height**	**Disease history**

{Prisoner's name and P_number } -> State, blood type, date of birth, weight, height and disease history are are full FD.

VISITS

Visitor ID	**P_number**	**Visitor Name**	**Visitor Phone number**	**Visitor Address**	**Duration Of visit**	**Time of visits**	**Number of visits**

{Visitor ID} -> P_number, Visitor Name, Visitor Phone number, Visitor Address, Duration of visit, Time of visits and Number of visits are full FD.

ACTIVITIES

Activity number	Reward	Activity type	P_number

{Activity number} -> Reward, and Activity type and P_number are full FD.

CELLS

Cell's number	Cell's capacity	Cell's state	F_number

{Cell's number} -> Cell's capacity and Cell's state and **F**_number are full FD.

WARDEN

Warden ID	First name	Last name	Gender	Address	Password	Salary

{Warden ID}-> Fname, Lname, Gender, Address, Password, and Salary are full FD.

STAFF

Staff member's ID	W_ID	First name	Last name	Gender	Job type	Salary	Address

PRISON COOK

ID

HEAD DOCTOR

ID

YARD OFFICERS

ID

PRISON OFFICES

ID	F_number

PRISONERS

Prisoner number	C_number	First name	Last name	Date of arrival	Date of release

HEALTH RECORDS

Prisoner's name	P_number	State	Blood type	Date of birth	Weight	Height	Disease history

CRIMINAL RECORDS

Prisoner's name	P_number	Prisoner's crime

VISITS

Visitor ID	P_number	Visitor Name	Visitor Phone number	Visitor Address	Duration Of visit	Time of visits	Number of visits

ACTIVITIES

Activity number	Activity type	Reward	P_number

MANAGED

P_number	S_ID

FLOOR

Floor number	Floor's type

CELL'S FLOOR

F_number	Number of cells

WARDENS FLOOR

F_number	Office number

CELLS

Cell's number	Cell's capacity	Cell's state	F_number

PUNISHMENT

Punishment number	W_ID	Punishment type

LAW VIOLATORS

Prisoner's name	Punish number

WARDEN

Warden ID	First name	Last name	Gender	Address	Password	Salary

4.3 Third Normal Form

-To make the relational schema in Third normal form [3NF], there should not be a Transitive dependency. Whenever some indirect relationship happens to cause functional dependency (FC), it is known as Transitive Dependency. Thus, if A -> B and B -> C are true, then A -> C happens to be a transitive dependency. Therefore, STAFF relation violates the 3NF guideline as explained below.

- **Staff members ID** -> **Job type** and **Job type** -> **Salary**.
- Since Job type is a non-prime attribute, and Salary is a non-prime attribute, hence the relation **STAFF** is not in 3NF.
- To normalize it, we decompose **STAFF** into the two 3NF relations:

STAFF

Staff member's Id	W_ID	First name	Last name	Gender	Job type	Salary	Address

JOB_SALARY

Job type	Salary

STAFF

Staff member's ID	W_ID	First name	Last name	Gender	Address	Job type

- **Activity number** -> **Activity type** and **Activity type** -> **Reward**
- Since Activity type is a non-prime attribute, and Reward is a non-prime attribute, hence the relation **ACTIVITIES** is not in 3NF.
- To normalize it, we decompose **ACTIVITIES** into the two 3NF relations:

ACTIVITIES

Activity number	Reward	Activity type	P_number

ACTIVITY_REWARD

Activity type	Reward

ACTIVITIES

Activity number	Activity type	P_number

STAFF

Staff member's ID	W_ID	First name	Last name	Gender	Job type	Address

JOB_SALARY

Job type	Salary

PRISON COOK

ID

HEAD DOCTOR

ID

YARD OFFICERS

ID

PRISON OFFICES

ID	F_number

PRISONERS

Prisoner number	C_number	First name	Last name	Date of arrival	Date of release

HEALTH RECORDS

Prisoner's name	P_number	State	Blood type	Date of birth	Weight	Height	Disease history

CRIMINAL RECORDS

Prisoner's name	P_number	Prisoner's crime

VISITS

Visitor ID	P_number	Visitor Name	Visitor Phone number	Visitor Address	Duration Of visit	Time of visits	Number of visits

ACTIVITIES

Activity number	Activity type	P_number

ACTIVITY_REWARD

Activity type	Reward

MANAGED

P_number	S_ID

FLOOR

Floor number	Floor's type

CELL'S FLOOR

F_number	Number of cells

WARDENS FLOOR

F_number	Office number

CELLS

Cell's number	Cell's capacity	Cell's state	F_number

PUNISHMENT

Punishment number	W_ID	Punishment type

LAW VIOLATORS

Prisoner's name	Punish number

WARDEN

Warden ID	First name	Last name	Gender	Address	Password	Salary

5 Final DB Schema Diagram

PART III: IMPLEMENTATION

6 Table Creation Script

6.1 < WARDEN > TABLE

CREATE TABLE WARDEN (

Warden_ID VARCHAR(30) NOT NULL PRIMARY KEY,

Fname varchar(15),

Lname varchar(15),

Gender varchar(20),

Address varchar(255),

Password NUMERIC(10),

Salary decimal(10,2) NOT NULL);

Query Editor Query History

```
CREATE TABLE WARDEN (
Warden_ID VARCHAR(30) NOT NULL PRIMARY KEY,
Fname varchar(15),
Lname varchar(15),
Gender varchar(20),
Address varchar(255),
Password NUMERIC(10),
Salary decimal(10,2) NOT NULL);
```

Not null constraint on salary since the Warden has to be paid.

6.2 < JOB_SALARY > TABLE

CREATE TABLE JOB_SALARY (

Job_Type VARCHAR(15) NOT NULL PRIMARY KEY,

Salary decimal(10,2) NOT NULL);

Query Editor Query History

```
CREATE TABLE JOB_SALARY (
Job_Type VARCHAR(15) NOT NULL PRIMARY KEY,
Salary decimal(10,2) NOT NULL);
```

Not null constraint on salary since all jobs have a salary.

6.3 < STAFF > TABLE

CREATE TABLE STAFF(

staffID VARCHAR(30) NOT NULL PRIMARY KEY,

W_ID VARCHAR(30) REFERENCES WARDEN (Warden_ID) ,

Fname VARCHAR(20),

Lname VARCHAR(20),

Gender VARCHAR(20),

Job_type VARCHAR(15) REFERENCES JOB_SALARY(Job_Type),

Address VARCHAR(255));

Query Editor Query History

```
CREATE TABLE STAFF(
staffID VARCHAR(30) NOT NULL PRIMARY KEY,
W_ID  VARCHAR(30) REFERENCES WARDEN (Warden_ID) ,
Fname    VARCHAR(20),
Lname    VARCHAR(20),
Gender   VARCHAR(20),
Job_type VARCHAR(15) REFERENCES JOB_SALARY(Job_Type),
Address  VARCHAR(255));
```

6.4 < FLOOR> TABLE

CREATE TABLE FLOOR (

Floor_number VARCHAR(30) NOT NULL PRIMARY KEY,

Floor's_type VARCHAR(30));

Notifications Query Editor Query History Data Output Mess

```
CREATE TABLE FLOOR (
Floor_number VARCHAR(30) NOT NULL PRIMARY KEY,
Floor's_type VARCHAR(30));
```

6.5 < PRISON_OFFICERS > TABLE

CREATE TABLE PRISON_OFFICERS(

ID VARCHAR(30) NOT NULL PRIMARY KEY,

F_number VARCHAR(30) REFERENCES FLOOR (Floor_number),

FOREIGN KEY (ID) REFERENCES STAFF (staffID) ON DELETE CASCADE);

Query Editor Query History

```
CREATE TABLE PRISON_OFFICERS(
ID VARCHAR(30) NOT NULL PRIMARY KEY,
F_number VARCHAR(30) REFERENCES FLOOR (Floor_number),
FOREIGN KEY (ID) REFERENCES STAFF (staffID) ON DELETE CASCADE);
```

6.6 < PRISON_COOK > TABLE

CREATE TABLE PRISON_COOK (

ID VARCHAR(30) NOT NULL PRIMARY KEY,

FOREIGN KEY (ID) REFERENCES STAFF (staffID) ON DELETE CASCADE);

Query Editor Query History

```
CREATE TABLE PRISON_COOK (
ID VARCHAR(30) NOT NULL PRIMARY KEY,
FOREIGN KEY (ID) REFERENCES STAFF (staffID) ON DELETE CASCADE);
```

6.7 < HAED_DOCTOR > TABLE

CREATE TABLE HEAD_DOCTOR(

ID VARCHAR(30) NOT NULL PRIMARY KEY,

FOREIGN KEY (ID) REFERENCES STAFF (staffID) ON DELETE CASCADE);

Query Editor Query History

```
CREATE TABLE HEAD_DOCTOR(
ID VARCHAR(30) NOT NULL PRIMARY KEY,
FOREIGN KEY (ID) REFERENCES STAFF (staffID) ON DELETE CASCADE);
```

6.8 < YARD_OFFICER > TABLE

CREATE TABLE YARD_OFFICERS (

ID VARCHAR(30) NOT NULL PRIMARY KEY,

FOREIGN KEY (ID) REFERENCES STAFF (staffID) ON DELETE CASCADE);

Query Editor Query History

```
CREATE TABLE YARD_OFFICERS (
ID VARCHAR(30) NOT NULL PRIMARY KEY,
FOREIGN KEY (ID) REFERENCES STAFF (staffID) ON DELETE CASCADE);
```

6.9 < CELLS > TABLE

CREATE TABLE CELLS(

Cells_number NUMERIC(10) NOT NULL PRIMARY KEY,

F_number VARCHAR(30) REFERENCES FLOOR(floor_number),

Cells_Capacity NUMERIC(10) DEFAULT 2,

Cells_state varchar(20));

Query Editor Query History

```
CREATE TABLE CELLS(
Cells_number NUMERIC(10) NOT NULL PRIMARY KEY,
F_number VARCHAR(30)  REFERENCES FLOOR(floor_number),
Cells_Capacity NUMERIC(10) DEFAULT 2,
Cells_state varchar(20) DEFAULT 'Unoccupied');
```

Since all cell capacities are the same, DEFAULT is used.

Default ‘unoccupied’ is used, until a prisoner occupies the cell.

6.10 < PRISONERS > TABLE

CREATE TABLE PRISONERS(

Prisoner_number NUMERIC(10) NOT NULL PRIMARY KEY,

Fname varchar(20),

Lname varchar(20),

Date_of_Arrival Date,

Date_of_Release Date,

C_number numeric(10),

FOREIGN KEY (C_number) REFERENCES CELLS (Cells_number) ON DELETE CASCADE);

Query Editor Query History

```
CREATE TABLE PRISONERS(
Prisoner_number NUMERIC(10) NOT NULL PRIMARY KEY,
Fname varchar(20),
Lname varchar(20),
Date_of_Arrival Date,
Date_of_Release Date,
C_number numeric(10),
FOREIGN KEY (C_number) REFERENCES CELLS (Cells_number) ON DELETE CASCADE);
```

6.11 < HEALTH_RECORDS > TABLE

CREATE TABLE HEALTH_RECORDS(

Prisoner_name varchar(50) NOT NULL,

P_number NUMERIC(10) NOT NULL,

State varchar(20),

Blood_type varchar(20),

Date_of_birth date,

Weight decimal(10,2),

Height decimal(10,2),

Disease_history varchar(255),

PRIMARY KEY(Prisoner_name, P_number),

FOREIGN KEY (P_number) REFERENCES PRISONERS (Prisoner_number) ON DELETE CASCADE);

Query Editor Query History

```
CREATE TABLE HEALTH_RECORDS(
Prisoner_name varchar(50) NOT NULL,
P_number NUMERIC(10) NOT NULL,
State varchar(20),
Blood_type varchar(20),
Date_of_birth date,
Weight decimal(10,2),
Height decimal(10,2),
Disease_history  varchar(255),
PRIMARY KEY(Prisoner_name, P_number),
FOREIGN KEY (P_number) REFERENCES PRISONERS (Prisoner_number) ON DELETE CASCADE);
```

(Prisoner_name) is partial key and (p_number) is a foreign key; those keys together will form the primary key for the weak entity.

6.12 < CRIMINAL_RECORDS > TABLE

CREATE TABLE CRIMINAL_RECORDS(

Prisoner_name varchar(50) NOT NULL ,

P_number NUMERIC(10) NOT NULL,

Prisoners_crime varchar(50) NOT NULL,

PRIMARY KEY(Prisoner_name, P_number),

FOREIGN KEY (P_number) REFERENCES PRISONERS (Prisoner_number) ON DELETE CASCADE);

Query Editor Query History

```
CREATE TABLE CRIMINAL_RECORDS(
Prisoner_name varchar(50) NOT NULL ,
P_number NUMERIC(10) NOT NULL,
Prisoners_crime varchar(50) NOT NULL,
PRIMARY KEY(Prisoner_name, P_number),
FOREIGN KEY (P_number) REFERENCES PRISONERS (Prisoner_number) ON DELETE CASCADE);
```

(Prisoner_name) is partial key and (p_number) is a foreign key; those keys together will form the primary key for the weak entity.

Since all prisoners have committed crimes to enter the jail, prisoners_crime must be NOT NULL.

6.13 < VISITS > TABLE

CREATE TABLE VISITS(

Visitor_ID VARCHAR(30) NOT NULL,

P_number NUMERIC(10),

Visitor_name varchar(50),

Visitor_phone_number NUMERIC(15),

Visitor_address varchar(50),

Duration_of_visit TIME NOT NULL,

Time_of_visit TIME NOT NULL,

Number_of_visits NUMERIC(10),

FOREIGN KEY (P_number) REFERENCES PRISONERS (Prisoner_number) ON DELETE CASCADE);

Query Editor Query History

```
CREATE TABLE VISITS(
Visitor_ID  VARCHAR(30) NOT NULL PRIMARY KEY,
P_number NUMERIC(10),
Visitor_name varchar(50),
Visitor_phone_number NUMERIC(15),
Visitor_address varchar(50),
Duration_of_visit TIME NOT NULL,
Time_of_visit TIME NOT NULL,
Number_of_visits NUMERIC(10),
FOREIGN KEY (P_number) REFERENCES PRISONERS (Prisoner_number) ON DELETE CASCADE);
```

Since all visitors are allowed to visit prisoners for a limited time, Time_of_visit and Duration_of_visit must be measured to not break the prison rules, therefore the two attributes will have the constraint NOT NULL.

6.14 < ACTIVITY_REWARD > TABLE

CREATE TABLE ACTIVITY_REWARD(

Activity_Type varchar(50) NOT NULL PRIMARY KEY,

Reward varchar(50));

Query Editor Query History

```
CREATE TABLE ACTIVITY_REWARD(
Activity_Type varchar(50) NOT NULL PRIMARY KEY,
Reward  varchar(50) );
```

6.15 < ACTIVITIES > TABLE

CREATE TABLE ACTIVITIES(

Activity_number NUMERIC(10) NOT NULL PRIMARY KEY,

Activity_Type varchar(50) REFERENCES ACTIVITY_REWARD (Activity_Type),

P_number NUMERIC(10) REFERENCES PRISONERS (Prisoner_number));

Query Editor Query History

```
CREATE TABLE ACTIVITIES(
Activity_number NUMERIC(10) NOT NULL PRIMARY KEY,
Activity_Type varchar(50) REFERENCES ACTIVITY_REWARD (Activity_Type),
P_number NUMERIC(10) REFERENCES PRISONERS (Prisoner_number) );
```

6.16 < MANAGED > TABLE

CREATE TABLE MANAGED (

P_number NUMERIC(10) NOT NULL,

S_ID varchar(30) NOT NULL,

PRIMARY KEY(P_number, S_ID) ,

FOREIGN KEY (P_number) REFERENCES PRISONERS (Prisoner_number)ON DELETE CASCADE,

FOREIGN KEY (S_ID) REFERENCES STAFF (staffID) ON DELETE CASCADE);

Query Editor Query History

```
CREATE TABLE MANAGED (
P_number NUMERIC(10) NOT NULL,
S_ID varchar(30) NOT NULL,
PRIMARY KEY(P_number, S_ID) ,
FOREIGN KEY (P_number) REFERENCES PRISONERS (Prisoner_number)ON DELETE CASCADE,
FOREIGN KEY (S_ID) REFERENCES STAFF (staffID) ON DELETE CASCADE );
```

6.17 < CELLS_FLOOR > TABLE

CREATE TABLE CELLS_FLOOR(

F_number VARCHAR(30) NOT NULL PRIMARY KEY,

NumberOFcells NUMERIC(100),

FOREIGN KEY (F_number) REFERENCES FLOOR (Floor_number) ON DELETE CASCADE);

Query Editor Query History

```
CREATE TABLE CELLS_FLOOR(
F_number VARCHAR(30) NOT NULL PRIMARY KEY,
NumberOFcells NUMERIC(100),
FOREIGN KEY (F_number) REFERENCES FLOOR (Floor_number) ON DELETE CASCADE );
```

6.18 < WARDEN_FLOOR > TABLE

CREATE TABLE WARDEN_FLOOR (

F_number VARCHAR(30) NOT NULL PRIMARY KEY,

Office_number NUMERIC(10),

FOREIGN KEY (F_number) REFERENCES FLOOR (Floor_number) ON DELETE CASCADE);

Query Editor Query History

```
CREATE TABLE WARDEN_FLOOR (
F_number VARCHAR(30) NOT NULL PRIMARY KEY,
Office_number NUMERIC(10),
FOREIGN KEY (F_number) REFERENCES FLOOR (Floor_number) ON DELETE CASCADE );
```

6.19 < PUNISHMENT > TABLE

CREATE TABLE PUNISHMENT (

Punishment_number NUMERIC(10) NOT NULL PRIMARY KEY,

W_ID VARCHAR(30) DEFAULT 200000,

Punishment_type varchar(255),

FOREIGN KEY (W_ID) REFERENCES Warden(Warden_ID) ON DELETE CASCADE);

Query Editor Query History

```
CREATE TABLE PUNISHMENT (
Punishment_number NUMERIC(10) NOT NULL PRIMARY KEY,
W_ID VARCHAR(30) DEFAULT 200000,
Punishment_type varchar(255),
FOREIGN KEY (W_ID) REFERENCES Warden(Warden_ID) ON DELETE CASCADE);
```

Since (W_ID) is the always the same, DEFAULT is used.

6.20 < LAW_VIOLATORS> TABLE

CREATE TABLE LAW_VIOLATERS (

Prisoners_name varchar(50) NOT NULL ,

Punishment_number NUMERIC(10) NOT NULL ,

PRIMARY KEY(Prisoners_name, Punishment_number),

FOREIGN KEY (Punishment_number) REFERENCES PUNISHMENT(Punishment_number) ON DELETE CASCADE);

Query Editor Query History

```
CREATE TABLE LAW_VIOLATERS (
Prisoners_name varchar(50) NOT NULL ,
Punishment_number NUMERIC(10) NOT NULL ,
PRIMARY KEY(Prisoners_name, Punishment_number),
FOREIGN KEY (Punishment_number) REFERENCES PUNISHMENT(Punishment_number) ON DELETE CASCADE);
```

(Prisoners_name) is partial key and (punishment_number) is a foreign key; those keys together will form the primary key for the weak entity.

7 Constraints Script

This section will show some of the constraints we've put in place in our database. Keep in mind that these aren't the only database restrictions. It is merely a representation of some of the most fundamental and significant ones.

Business Rule	SQL Script	Table
Each staff member has a unique ID	Implementation: while creating the STAFF table staffID VARCHAR(30) **NOT NULL PRIMARY KEY**	STAFF
Subclasses of STAFF must have an inherited ID. Since they are specializations of the superclass STAFF, if a STAFF member is deleted the data of all the referencing relations are deleted as well	FOREIGN KEY (ID) REFERENCES STAFF (staffID) **ON DELETE CASCADE**);	PRISON_OFFICERS YARD_OFFICERS HEAD_DOCTOR PRISON_COOK
Each CRIMINAL RECORD has the PRISONER'S crime	Prisoners_crime varchar(50) **NOT NULL**	CRIMINAL_RECORDS
Each Cell that has fewer prisoners in them than the cell's capacity is given a default label of "unoccupied"	Cells_state varchar(20) **DEFAULT 'Unoccupied'**);	CELLS

8 Queries and Transactions

The majority of our important queries and transactions do not need to be complicated. However, the system may require some more complicated queries and transactions.

8.1 <Average salaries >

Query in Natural Language (English)

This query will retrieve the average salary of each job type with an average salary greater than the specified department average salary.

Our database will be used in this context by users to get the average salary of each essential job type for the STAFF members (Prison officers, yard officers, head doctor, prison cook).

SQL Script

SELECT AVG(salary) AS "Ave Salary", job_type

FROM job_salary

Group by job_type

HAVING AVG(salary) > (select avg(salary)

FROM job_salary

Where job_type='YardOfficer');

Query Editor Query History

```
SELECT AVG(salary) AS "Ave Salary", job_type
FROM job_salary
Group by job_type
HAVING AVG(salary) > (select avg(salary)
FROM job_salary
Where job_type='YardOfficer');
```

Caption of the First Five Rows of the Output

Data Output Explain Messages Notifications

	Avg Salary numeric	job_type [PK] character varying (15)
1	12000.0000000000000000	PrisonOfficer
2	15000.0000000000000000	HeadDoctor

8.2 *<Specify prisoners with certain characteristics>*

Query in Natural Language (English)

This query will display the Prisoner number of all prisoners who will be released after the date

(1-1-2026), and their initials are not the letter (R).

SQL Script

select PRISONERS.Prisoner_number

from PRISONERS

where Date_of_Release > '1-1-2026' AND Fname NOT LIKE 'R%';

Query Editor Query History

```
1 select PRISONERS.Prisoner_number
2 from PRISONERS
3 where Date_of_Release > '1-1-2026' AND Fname NOT LIKE 'R%';
```

Caption of the First Five Rows of the Output

	prisoner_number [PK] numeric (10)
1	3
2	10
3	11
4	12
5	13
6	17
7	20
8	21
9	22
10	24

8.3 *<Specify robbers>*

Query in Natural Language (English)

Print ID, first name, and last name of prisoners who have committed robbery.

SQL Script

Select PRISONERS.Prisoner_number, Fname,Lname

From PRISONERS JOIN CRIMINAL_RECORDS ON Prisoner_number = P_number

Where PRISONERS.Prisoner_number in(select P_number

From CRIMINAL_RECORDS

Where Prisoners_crime LIKE 'Robbery');

Query Editor Query History

```
Select PRISONERS.Prisoner_number, Fname,Lname
From PRISONERS JOIN CRIMINAL_RECORDS ON Prisoner_number = P_number
Where PRISONERS.Prisoner_number in( select P_number
From CRIMINAL_RECORDS
Where Prisoners_crime  LIKE 'Robbery');
```

Caption of the First Five Rows of the Output

	prisoner_number [PK] numeric (10)	fname character varying (20)	lname character varying (20)
1	1	Kholood	Alghamdi
2	4	Khadija	Turkestani
3	5	Raana	Bukalalkri

8.4 *<Salary Raise>*

Query in Natural Language (English)

This query will raise the salary of all employees by 10%.

SQL Script

SELECT job_type, Salary, Salary*1.10 AS Salary_After_Bonus

FROM JOB_SALARY;

Query Editor Query History

```
SELECT job_type, Salary, Salary*1.10 AS Salary_After_Bonus

FROM JOB_SALARY;
```

Caption of the First Five Rows of the Output

Data Output Explain Messages Notifications

	job_type [PK] character varying (15)	salary numeric (10,2)	salary_after_bonus numeric
1	HeadDoctor	15000.00	16500.0000
2	YardOfficer	10000.00	11000.0000
3	PrisonCook	10000.00	11000.0000
4	PrisonOfficer	12000.00	13200.0000

8.5 *<Specify jobs with less than two members >*

Query in Natural Language (English)

This query will display jobs that have less than 2 members working in them including their names.

SQL Script

SELECT Fname,Lname, Job_type

FROM STAFF WHERE Job_type IN

(SELECT Job_type FROM STAFF

GROUP BY Job_type HAVING COUNT(*)<2);

Query Editor Query History

```
1  SELECT Fname,Lname, Job_type
2  FROM STAFF WHERE  Job_type IN
3  (SELECT Job_type FROM STAFF
4  GROUP BY Job_type HAVING COUNT(*)<2);
```

Caption of the First Five Rows of the Output

	fname character varying (20)	lname character varying (20)	job_type character varying (15)
1	Shahad	Balkhasher	PrisonCook
2	Wajd	Alharbi	HeadDoctor

8.6<*Total Number of Staff Members Per Job Type*>

Query in Natural Language (English)

This query will display names of the jobs and the numbers of staff memebrs working in descending order. We used the aggregate function COUNT here to find the total number of members working in each job type.

SQL Script

SELECT job_type, COUNT(*) AS NUMBER_OF_STAFF_PER_JOB_TYPE

FROM STAFF

GROUP BY Job_type

ORDER BY COUNT(*) DESC;

Query Editor Query History

```
SELECT job_type, COUNT(*) AS NUMBER_OF_STAFF_PER_JOB_TYPE
FROM STAFF
GROUP BY Job_type
ORDER BY COUNT(*) DESC;
```

Caption of the First Five Rows of the Output

	job_type character varying (15)	number_of_staff_per_job_type bigint
1	PrisonOfficer	20
2	YardOfficer	10
3	HeadDoctor	1
4	PrisonCook	1

8.7<*The Largest Number of Staff Members in a Job Type*>

Query in Natural Language (English)

This query will display the largest number of staff members that are working in a job among all job types. We used nested aggregate functions (MAX, COUNT).

SQL Script

SELECT MAX(NUMBER_OF_STAFF_PER_JOB_TYPE) AS "The Largest Number of Staff Members in a Job Type"

FROM(SELECT Job_type, COUNT(*)AS NUMBER_OF_STAFF_PER_JOB_TYPE

FROM STAFF

GROUP BY Job_type)

AS SubQueryAlias;

Query Editor Query History

```
SELECT MAX(NUMBER_OF_STAFF_PER_JOB_TYPE) AS "The Largest Number of Staff Members in a Job Type"
FROM(SELECT Job_type, COUNT(*)AS NUMBER_OF_STAFF_PER_JOB_TYPE
FROM STAFF
GROUP BY Job_type)
AS SubQueryAlias;
```

Caption of the First Five Rows of the Output

	The Largest Number of Staff Members in a Job Type bigint
1	20

8.8 Update Example

8.8.1 example

Update in Natural Language (English)

UPDATE the phone number of the visitor with ID (10015).

SQL Script

Query Editor Query History

```sql
UPDATE visits
SET visitor_phone_number = '0505355475'
WHERE visitor_id = '10015';
```

Before Update:

Data Output Explain Messages Notifications

	visitor_id [PK] character varying (30)	p_number numeric (10)	visitor_name character varying (50)	visitor_phone_number numeric (15)	visitor_address character varying (50)	duration_of_visit time without time zone	time_of_visit time without time zone	number_of_visits numeric (10)
1	10015	8	[...] Batool Ghanam	546621179	[...] Jeddah	01:00:00	09:00:00	1
2	10028	20	[...] Lamia Sabbagh	561478861	[...] Riyadh	00:40:00	07:20:00	4
3	10113	16	[...] Amjad Alharthi	545100322	[...] Jeddah	00:50:00	13:10:00	1
4	10184	23	[...] Sumayah Hajj	556280600	[...] Jeddah	00:50:00	07:00:00	2
5	10204	10	Haneen Hejazi	500198917	[...] Yanbu	00:30:00	14:00:00	3
6	10232	19	[...] Raneem Halawani	503378190	[...] Jeddah	00:40:00	07:30:00	1
7	10267	8	[...] Mawada Alsabban	562323446	[...] Makkah	01:00:00	10:15:00	3
8	10288	6	[...] Sultana Mukhtar	572133320	[...] Jeddah	00:20:00	12:50:00	1
9	10336	11	[...] Sondos Bawazeer	572100118	[...] Dammam	01:00:00	13:10:00	1
10	10503	4	[...] Rahaf Algasem	558070977	[...] Hail	00:40:00	13:30:00	3
11	10535	13	[...] Dania Farsi	541085426	[...] Taif	00:20:00	08:40:00	1
12	10570	17	[...] Sana Alhadi	542120066	[...] Jeddah	00:20:00	11:00:00	1
13	10810	2	[...] Fatima Alhakami	546055321	[...] Jeddah	00:30:00	10:30:00	2
14	10834	4	[...] Heba Bagader	550123781	[...] Tabouk	00:30:00	10:30:00	1

After Update:

	visitor_id [PK] character varying (30)	p_number numeric (10)	visitor_name character varying (50)	visitor_phone_number numeric (15)	visitor_address character varying (50)	duration_of_visit time without time zone	time_of_visit time without time zone	number_of_visits numeric (10)
1	10015	8	[...] Batool Ghanam	505355475	[...] Jeddah	01:00:00	09:00:00	1
2	10028	20	[...] Lamia Sabbagh	561478861	[...] Riyadh	00:40:00	07:20:00	4
3	10113	16	[...] Amjad Alharthi	545100322	[...] Jeddah	00:50:00	13:10:00	1
4	10184	23	[...] Sumayah Hajj	556280600	[...] Jeddah	00:50:00	07:00:00	2
5	10204	10	Haneen Hejazi	500198917	[...] Yanbu	00:30:00	14:00:00	3
6	10232	19	[...] Raneem Halawani	503378190	[...] Jeddah	00:40:00	07:30:00	1
7	10267	8	[...] Mawada Alsabban	562323446	[...] Makkah	01:00:00	10:15:00	3
8	10288	6	[...] Sultana Mukhtar	572133320	[...] Jeddah	00:20:00	12:50:00	1
9	10336	11	[...] Sondos Bawazeer	572100118	[...] Dammam	01:00:00	13:10:00	1
10	10503	4	[...] Rahaf Algasem	558070977	[...] Hail	00:40:00	13:30:00	3
11	10535	13	[...] Dania Farsi	541085426	[...] Taif	00:20:00	08:40:00	1
12	10570	17	[...] Sana Alhadi	542120066	[...] Jeddah	00:20:00	11:00:00	1
13	10810	2	[...] Fatima Alhakami	546055321	[...] Jeddah	00:30:00	10:30:00	2
14	10834	4	[...] Heba Bagader	550123781	[...] Tabouk	00:30:00	10:30:00	1

8.8.2 example

Update in Natural Language (English)

Update the cell's capacity of the cell with number 30 from 2 to 1.

SQL Script

```
UPDATE CELLS
SET Cells_Capacity = 1
WHERE Cells_number= 30;
```

Caption of the Output before update:

	cells_number [PK] numeric (10)	f_number character varying (30)	cells_capacity numeric (10)	cells_state character varying (20)
27	27	3	2	Unoccupied
28	28	3	2	Unoccupied
29	29	3	2	Unoccupied
30	30	3	2	Unoccupied

Caption of the Output after update:

	cells_number [PK] numeric (10)	f_number character varying (30)	cells_capacity numeric (10)	cells_state character varying (20)
27	27	3	2	Unoccupied
28	28	3	2	Unoccupied
29	29	3	2	Unoccupied
30	30	3	1	Unoccupied

8.9 Delete Example

8.9.1 example

Delete in Natural Language (English)

Delete the prisoners with the number (00025).

SQL Script

```sql
DELETE FROM Prisoners WHERE prisoner_number='00025';
```

Caption of the Output before delete:

	prisoner_number [PK] numeric (10)	fname character varying (20)	lname character varying (20)	date_of_arrival date	date_of_release date	c_number numeric (10)
19	19	Malak	Ramadan	2017-06-03	2022-04-01	10
20	20	Jouri	Nasser	2018-04-01	2050-08-02	10
21	21	Joud	Noumani	2020-01-01	2044-06-01	11
22	22	Jood	Almokri	2012-12-12	2042-02-03	11
23	23	Fajr	Alahmadi	2009-04-01	2010-01-18	12
24	24	Suha	Alqahtani	2004-11-14	2045-09-01	12
25	25	Samar	Alharthi	2008-03-01	2011-05-15	13

Caption of the Output after delete:

Data Output | Explain | Messages | Notifications

	prisoner_number [PK] numeric (10)	fname character varying (20)	lname character varying (20)	date_of_arrival date	date_of_release date	c_number numeric (10)
19	19	Malak	Ramadan	2017-06-03	2022-04-01	10
20	20	Jouri	Nasser	2018-04-01	2050-08-02	10
21	21	Joud	Noumani	2020-01-01	2044-06-01	11
22	22	Jood	Almokri	2012-12-12	2042-02-03	11
23	23	Fajr	Alahmadi	2009-04-01	2010-01-18	12
24	24	Suha	Alqahtani	2004-11-14	2045-09-01	12

8.9.2 example

Delete in Natural Language (English)

Delete the STAFF member with the ID (20021).

SQL Script

```
1  DELETE FROM STAFF WHERE staffID='20021';
```

Caption of the Output before delete:

	staffid [PK] character varying (30)	w_id character varying (30)	fname character varying (20)	lname character varying (20)	gender character varying (20)	job_type character varying (15)	address character varying (255)
1	20001	200000	Fai	Alnagrani	Female	YardOfficer	[...] Jeddah
2	20002	200000	Lina	Alghamdi	Female	YardOfficer	Riyadh
3	20003	200000	Shahad	Balkhasher	Female	PrisonCook	Jeddah
4	20004	200000	Taleen	Balkasher	Female	YardOfficer	Jeddah
5	20005	200000	Tamara	Bukhari	Female	YardOfficer	Dammam

Caption of the Output after delete:

	staffid [PK] character varying (30)	w_id character varying (30)	fname character varying (20)	lname character varying (20)	gender character varying (20)	job_type character varying (15)	address character varying (255)
17	20017	200000	Lana	Kurdi	Female	PrisonOfficer	Hail
18	20018	200000	Sara	khatib	Female	PrisonOfficer	Taif
19	20019	200000	Renad	katib	Female	PrisonOfficer	Tabouk
20	20020	200000	Jamila	Khrgi	Female	PrisonOfficer	Abha

9 APPENDIX

salary
password
address
warden ID
gender
WARDEN
Name
First Name
Last Name
punishment number
punishment type
PUNISHMENT
DISCIPLINES
LAW VIOLATORS
prisoner's names
DECIDES
floor's type
floor number
OVERSEES
SURVELLANCED
FLOOR
warden's floor
office number
Job_type
Gender
Staff member's ID
Salary
Address
PRISON OFFICER
cell's floor
number of cells
STAFF
YARD OFFICER
OBSERVE
PRISON YARD
HEAD DOCTOR
HEADS
HEALTH CARE
PRISON COOK
WORKS IN
CANTEEN
CONTAINS
State
Disease history
Blood type
Name
HEALTH RECORDS
Date of birth
Weight
Height
Has
MANAGED
Prisoner's number
Date of Arrival
PRISONER
Date of Realease
PLACED
CELLS
cell's number
cell's capacity
cell's state
ASSIGNED TO
RECIEVE
OWNS
prisoner's names
prisoner's crimes
CRIMINAL RECORDS
VISITS
number of visits
time of visits
duration of visit
visitor phone number
visitor address
visitor ID
visitor name
ACTIVITIES
activity type
Activity number
reward

WARDEN TABLE

	warden_id [PK] character varying (30)	fname character varying (15)	lname character varying (15)	gender character varying (20)	address character varying (255)	password numeric (10)	salary numeric (10,2)
1	200000	Majd	Alzahrani	Female	Jeddah	1580	50000.00

JOB_SALARY TABLE

	job_type [PK] character varying (15)	salary numeric (10,2)
1	HeadDoctor	15000.00
2	PrisonCook	10000.00
3	PrisonOfficer	12000.00
4	YardOfficer	10000.00

STAFF TABLE

	staffid [PK] character varying (30)	w_id character varying (30)	fname character varying (20)	lname character varying (20)	gender character varying (20)	job_type character varying (15)	address character varying (255)
1	20001	200000	Fai	Alnagrani	Female	YardOfficer	[...] Jeddah
2	20002	200000	Lina	Alghamdi	Female	YardOfficer	Riyadh
3	20003	200000	Shahad	Balkhasher	Female	PrisonCook	Jeddah
4	20004	200000	Taleen	Balkasher	Female	YardOfficer	Jeddah
5	20005	200000	Tamara	Bukhari	Female	YardOfficer	Dammam
6	20006	200000	Joud	Alsindi	Female	YardOfficer	Madina
7	20007	200000	Ahlam	Almalki	Female	YardOfficer	Makkah
8	20008	200000	Wajd	Alharbi	Female	HeadDoctor	Jeddah
9	20009	200000	Talia	Abualrrab	Female	YardOfficer	Riyadh
10	20010	200000	Ghazal	Jan	Female	YardOfficer	Jeddah
11	20011	200000	Sedrh	Algihani	Female	YardOfficer	Jeddah
12	20012	200000	Sara	Alshamri	Female	YardOfficer	Makkah
13	20013	200000	Maysoon	Talhan	Female	PrisonOfficer	Makkah
14	20014	200000	Monera	Algahtani	Female	PrisonOfficer	Jeddah
15	20015	200000	Reem	Aljahdali	Female	PrisonOfficer	Jeddah
16	20016	200000	Maya	Abutaleb	Female	PrisonOfficer	Madina
17	20017	200000	Lana	Kurdi	Female	PrisonOfficer	Hail
18	20018	200000	Sara	khatib	Female	PrisonOfficer	Taif
19	20019	200000	Renad	katib	Female	PrisonOfficer	Tabouk
20	20020	200000	Jamila	Khrgi	Female	PrisonOfficer	Abha
21	20021	200000	Reem	Alshahri	Female	PrisonOfficer	Jeddah
22	20022	200000	Nadia	Almutairi	Female	PrisonOfficer	Jeddah
23	20023	200000	Maria	Albakri	Female	PrisonOfficer	Makkah
24	20024	200000	Lana	Talaat	Female	PrisonOfficer	Jeddah
25	20025	200000	Zainah	Aljahdali	Female	PrisonOfficer	Jeddah
26	20026	200000	Sara	Kazim	Female	PrisonOfficer	Madina
27	20027	200000	Fatima	Ahmed	Female	PrisonOfficer	Hail
28	20028	200000	Sheikhah	Mohammed	Female	PrisonOfficer	Taif
29	20029	200000	Noor	Algabasani	Female	PrisonOfficer	Tabouk
30	20030	200000	Aisha	Almalki	Female	PrisonOfficer	Abha
31	20031	200000	Mhra	Bkari	Female	PrisonOfficer	Jeddah
32	20032	200000	Nojood	Alghamdi	Female	PrisonOfficer	Jeddah

FLOOR TABLE

	floor_number [PK] character varying (30)	floor's_type character varying (30)
1	1	Cells Floor
2	2	Cells Floor
3	3	Cells Floor
4	4	Cells Floor
5	5	Cells Floor
6	6	Cells Floor
7	7	Cells Floor
8	8	Cells Floor
9	9	Cells Floor
10	10	Cells Floor
11	11	Warden Floor

PRISON_OFFICERS TABLE

	id [PK] character varying (30)	f_number character varying (30)
1	20013	1
2	20014	1
3	20015	2
4	20016	2
5	20017	3
6	20018	3
7	20019	4
8	20020	4
9	20021	5
10	20022	5
11	20023	6
12	20024	6
13	20025	7
14	20026	7
15	20027	8
16	20028	8
17	20029	9
18	20030	9
19	20031	10
20	20032	10

PRISON_COOK TABLE

	id [PK] character varying (30)
1	20003

HEAD_DOCTOR TABLE

	id [PK] character varying (30)
1	20008

YARD_OFFICER TABLE

	id [PK] character varying (30)
1	20001
2	20002
3	20004
4	20005
5	20006
6	20007
7	20009
8	20010
9	20011
10	20012

CELLS TABLE

	cells_number [PK] numeric (10)	f_number character varying (30)	cells_capacity numeric (10)	cells_state character varying (20)
1	1	1	2	Occupied
2	2	1	2	Occupied
3	3	1	2	Occupied
4	4	1	2	Occupied
5	5	1	2	Occupied
6	6	2	2	Occupied
7	7	2	2	Occupied
8	8	2	2	Occupied
9	9	2	2	Occupied
10	10	2	2	Occupied
11	11	3	2	Occupied
12	12	3	2	Occupied
13	13	3	2	Unoccupied
14	14	3	2	Unoccupied
15	15	3	2	Unoccupied
16	16	4	2	Unoccupied
17	17	4	2	Unoccupied
18	18	4	2	Unoccupied
19	19	4	2	Unoccupied
20	20	4	2	Unoccupied
21	21	5	2	Unoccupied
22	22	5	2	Unoccupied
23	23	5	2	Unoccupied
24	24	5	2	Unoccupied
25	25	5	2	Unoccupied
26	26	6	2	Unoccupied
27	27	6	2	Unoccupied
28	28	6	2	Unoccupied
29	29	6	2	Unoccupied
30	30	6	2	Unoccupied

31	31	7	2	Unoccupied
32	32	7	2	Unoccupied
33	33	7	2	Unoccupied
34	34	7	2	Unoccupied
35	35	7	2	Unoccupied
36	36	8	2	Unoccupied
37	37	8	2	Unoccupied
38	38	8	2	Unoccupied
39	39	8	2	Unoccupied
40	40	8	2	Unoccupied
41	41	9	2	Unoccupied
42	42	9	2	Unoccupied
43	43	9	2	Unoccupied
44	44	9	2	Unoccupied
45	45	9	2	Unoccupied
46	46	10	2	Unoccupied
47	47	10	2	Unoccupied
48	48	10	2	Unoccupied
49	49	10	2	Unoccupied
50	50	10	2	Unoccupied

PRISONERS TABLE

	prisoner_number [PK] numeric (10)	fname character varying (20)	lname character varying (20)	date_of_arrival date	date_of_release date	c_number numeric (10)
1	1	Kholood	Aghamdi	2012-01-01	2014-01-01	1
2	2	Arein	Malki	2001-07-09	2021-05-01	1
3	3	Abeer	Baisa	2003-02-03	2026-01-04	2
4	4	Khadija	Turkestani	1995-01-01	1999-01-01	2
5	5	Raana	Bukalalkri	2008-08-15	2012-08-19	3
6	6	Dhai	Kassar	2006-07-04	2016-08-07	3
7	7	Tahani	Aghamdi	1990-12-10	1996-01-20	4
8	8	Raghad	Alharbi	1998-03-15	2038-03-30	4
9	9	Taleen	Alselami	2001-07-16	2006-05-13	5
10	10	Tala	Alasadi	1999-06-11	2032-03-01	5
11	11	Marwa	Sendi	2008-03-12	2040-09-14	6
12	12	Amal	Altoweejri	2005-02-01	2050-07-01	6
13	13	Sara	Ghahtani	2003-01-01	2035-01-14	7
14	14	Nadeen	Abutteen	2012-09-30	2022-08-07	7
15	15	Rawan	BinSaad	2019-08-17	2039-12-01	8
16	16	Layaly	Aldosari	2019-05-09	2022-08-01	8
17	17	Ahlam	Binmahfouz	2020-05-01	2037-09-01	9
18	18	Shatha	Hashmi	2006-05-01	2022-12-28	9
19	19	Malak	Ramadan	2017-06-03	2022-04-01	10
20	20	Jouri	Nasser	2018-04-01	2050-08-02	10
21	21	Joud	Noumani	2020-01-01	2044-06-01	11
22	22	Jood	Almokri	2012-12-12	2042-02-03	11
23	23	Fajr	Alahmadi	2009-04-01	2010-01-18	12
24	24	Suha	Alqahtani	2004-11-14	2045-09-01	12
25	25	Samar	Alharthi	2008-03-01	2011-05-15	13

HEALTH_RECORDS TABLE

	prisoner_name [PK] character varying (5	p_number [PK] numeric (10)	state character varying (20)	blood_type character varying (20)	date_of_birth date	weight numeric (10,2)	height numeric (10,2)	disease_history character varying (255)
1	Abeer Baisa	3	Single	[...] O-	1978-11-15	110.20	159.40	[...] Diabetic and has kidney stones
2	Ahlam Binmahfouz	17	Widowed	B-	2000-04-23	65.30	170.50	[null]
3	Amal Altoweejri	12	Single	AB+	1976-02-17	76.40	165.60	Mumps
4	Arein Malki	2	divorced	[...] B+	1980-12-03	115.80	178.30	[...] Has E. coli and low blood pressure
5	Dhai Kassar	6	Single	[...] AB+	1970-08-09	52.10	167.30	[...] chicken pox
6	Fajr Alahmadi	23	Single	B-	1989-10-05	56.30	172.40	E. Coli and an allergy to cotton
7	Jood Almokri	22	Divorced	O+	1981-09-05	82.60	180.90	[null]
8	Joud Noumani	21	[...] Single	[...] A+	1990-08-05	95.10	154.30	Diabetic and has heart problems
9	Jouri Nasser	20	Widowed	B+	1992-07-05	55.00	150.50	Has Kidney stones and mild migraines
10	Khadija Turkestani	4	[...] married	[...] O+	1960-10-14	82.50	167.20	[...] Pneumonia and respiratory complications
11	Kholood Aghamdi	1	[...] Single	[...] A-	1979-07-04	82.10	168.30	[...] diabetic and allergic to nuts
12	Layaly Aldosari	16	Single	[...] O+	1994-03-20	58.30	156.20	[...] Diabetic and anemic
13	Malak Ramadan	19	Single	AB+	1978-06-15	84.80	178.50	Diabetic and allergic to fish
14	Marwa Sendi	11	[...] widowed	[...] O-	1979-03-06	82.10	183.30	[null]
15	Nadeen Abutteen	14	Married	B-	1989-01-03	49.50	162.70	[null]
16	Raghad Alharbi	8	married	[...] AB+	1975-06-17	64.00	173.30	[...] Diabetic and allergic to fish
17	Rana Bukalalkri	5	[...] widowed	[...] B+	1969-09-25	94.80	183.20	[...] Rubella and asthma
18	Rawan BinSaad	15	Widowed	O-	1986-02-07	93.10	185.50	pneumonia and asthma
19	Samar Alharthi	25	Widowed	AB+	1189-12-01	70.30	168.50	[null]
20	Sara Ghahtani	13	Divorced	A+	1972-01-05	79.80	167.20	Malaria
21	Shatha Hashmi	18	Married	AB+	2001-05-30	89.30	181.90	Lyme disease and anemia
22	Suha Alqahtani	24	Divorced	A-	1985-11-21	105.40	175.60	Rubella, severe migraines, and asthma
23	Tahani Aghamdi	7	[...] divorced	[...] A-	1969-07-06	72.50	193.20	[...] Diabetic and Anemic
24	Tala Alasadi	10	[...] Single	[...] B+	1976-04-10	84.20	178.30	[...] Measles and occasional migraines
25	Taleen Alselami	9	[...] Single	[...] B-	1981-05-12	63.60	158.00	[...] HIV positive

CRIMINAL_RECORDS TABLE

	prisoner_name [PK] character varying (50)	p_number [PK] numeric (10)	prisoners_crime character varying (50)
1	Abeer Baisa	3	Homicide
2	Ahlam Binmahfouz	17	kidnapping
3	Amal Altoweejri	12	Human Trafficking
4	Arein Malki	2	Drug Possession
5	Dhai Kassar	6	Drug Possession
6	Fajr Alahmadi	23	Driving Under the influence
7	Jood Almokri	22	Murder
8	Joud Noumani	21	Homicide
9	Jouri Nasser	20	Drug Possession
10	Khadija Turkestani	4	Robbery
11	Kholood Aghamdi	1	Robbery
12	Layaly Aldosari	16	Car Theft
13	Malak Ramadan	19	Car Theft
14	Marwa Sendi	11	Homicide
15	Nadeen Abutteen	14	Cyber Extortion
16	Raana Bukalalkri	5	Robbery
17	Raghad Alharbi	8	Terrorism
18	Rawan BinSaad	15	Bank Robbery
19	Samar Alharthi	25	Sexual Assault
20	Sara Ghahtani	13	kidnapping
21	Shatha Hashmi	18	Aggravated Assault
22	Suha Alqahtani	24	Homicide
23	Tahani Aghamdi	7	Gun Possession
24	Tala Alasadi	10	Homicide
25	Taleen Alselami	9	Gun Possession

VISITS TABLE

	visitor_id [PK] character varying (30)	p_number numeric (10)	visitor_name character varying (50)	visitor_phone_number numeric (15)	visitor_address character varying (50)	duration_of_visit time without time zone	time_of_visit time without time zone	number_of_visits numeric (10)
1	10015	8	Batool Ghanam	546621179	Jeddah	01:00:00	09:00:00	1
2	10028	20	Lamia Sabbagh	561478861	Riyadh	00:40:00	07:20:00	4
3	10113	16	Amjad Alharthi	5445100322	Jeddah	00:50:00	13:10:00	1
4	10128	25	Reem Falmban	552100767	Madina	00:30:00	11:10:00	1
5	10184	23	Sumayah Hajj	556280600	Jeddah	00:50:00	07:00:00	2
6	10204	10	Haneen Hejazi	500198917	Yanbu	00:30:00	14:00:00	3
7	10232	19	Raneem Halawani	503378190	Jeddah	00:40:00	07:30:00	1
8	10267	8	Mawada Alsabban	562323446	Makkah	01:00:00	10:15:00	3
9	10288	6	Sultana Mukhtar	572133320	Jeddah	00:20:00	12:50:00	1
10	10336	11	Sondos Bawazeer	572100118	Dammam	01:00:00	13:10:00	1
11	10503	4	Rahaf Algasem	558070977	Hail	00:40:00	13:30:00	3
12	10535	13	Dania Farsi	541085426	Taif	00:20:00	08:40:00	1
13	10570	17	Sana Alhadi	5442120066	Jeddah	00:20:00	11:00:00	1
14	10810	2	Fatima Alhakami	546055321	Jeddah	00:30:00	10:30:00	2
15	10834	4	Heba Bagader	550123781	Tabouk	00:30:00	10:30:00	1

ACTIVITY_REWARD TABLE

	activity_type [PK] character varying (50)	reward character varying (50)
1	Cleaner	More time out of the cell
2	Community Work	Reduced Sentence
3	Electrician	Access to TVs and games consoles
4	Farmer	Limited Internet Access
5	Food Service	[...] Financial reward
6	Groundskeeper	Improve living conditions
7	Laundry	Financial Reward
8	Librarian	Financial Reward
9	Plumber	More time out of the cell
10	Woodwork	Limited Internet Access

ACTIVITIES TABLE

	activity_number [PK] numeric (10)	activity_type character varying (50)	p_number numeric (10)
1	1	Cleaner	4
2	2	Laundry	22
3	3	Community Work	17
4	4	Librarian	11
5	5	Farmer	9
6	6	Woodwork	2
7	7	Food Service	24
8	8	Plumber	15
9	9	Electrician	12
10	10	Groundskeeper	19

MANAGED TABLE

	p_number [PK] numeric (10)	s_id [PK] character varying (30)
1	1	20001
2	1	20002
3	1	20003
4	1	20004
5	1	20005
6	1	20006
7	1	20007
8	1	20008
9	1	20009
10	1	20010
11	1	20011
12	1	20012
13	1	20013
14	1	20014
15	2	20001
16	2	20002
17	2	20003
18	2	20004
19	2	20005
20	2	20006
21	2	20007
22	2	20008
23	2	20009
24	2	20010
25	2	20011
26	2	20012
27	2	20013
28	2	20014
29	3	20001
30	3	20002
31	3	20003
32	3	20004
33	3	20005
34	3	20006
35	3	20007
36	3	20008
37	3	20009
38	3	20010
39	3	20011
40	3	20012
41	3	20015
42	3	20016

And so on.

CELLS_FLOOR TABLE

	f_number [PK] character varying (30)	numberofcells numeric (100)
1	1	5
2	2	5
3	3	5
4	4	5
5	5	5
6	6	5
7	7	5
8	8	5
9	9	5
10	10	5

WARDEN_FLOOR TABLE

	f_number [PK] character varying (30)	office_number numeric (10)
1	11	1

PUNISHMENT TABLE

	punishment_number [PK] numeric (10)	w_id character varying (30)	punishment_type character varying (255)
1	1	200000	Electric chair
2	2	200000	Punitive Segregation
3	3	200000	Extending Sentences
4	4	200000	Water Deprivation
5	5	200000	Losing Visitation Privilege
6	6	200000	corporal punishment
7	7	200000	[...] Withdrawing the right to buy items from the commissary
8	8	200000	Transfer to another higher-security prison
9	9	200000	Incapacitation
10	10	200000	Rehabilitation

LAW_VIOLATORS TABLE

	prisoners_name [PK] character varying (50)	punishment_number [PK] numeric (10)
1	Abeer Baisa	4
2	Jood Almokri	7
3	Jouri Nasser	6
4	Kholood Alghamdi	5
5	Marwa Sendi	9
6	Raana Bukalalkri	1
7	Raghad Alharbi	10
8	Samar Alharthi	3
9	Sara Alqahtani	3
10	Shatha Hashmi	2
11	Suha Alqahtani	8
12	Taleen Alselami	5