Members: Jax Fonseca-Folden, Vimal Murugan, Jean Carlos Andrade

Sprint 3

REQUIREMENTS

Refine the user stories that you made in previous sprint. List your updated user stories in decreasing order of priority. Highlight the stories that were completed in Sprint 1 in one color. Highlight the stories that were completed in Sprint 2 in a different color. Highlight the updated/new stories chosen for Sprint 3, if any, in a third color. *There is no need to explicitly show your story refinement process.* Use the format shown below.

Story ID	Story description
US 01	As a <Crime analyst>, I research crime trends over time.
US 02	As an <Investigator>, I investigate the circumstances of a specific case.
US 03	As an <Investigator>, I investigate suspects.
US 04	As an <Investigator>, I find evidence to prosecute suspects.
US 05	As an <Investigator>, I arrest suspects.
US 06	As a <Lawyer>, I represent my client in court.
US 07	As a <Lawyer>, I construct a narrative for the party I am representing.
US 08	As a <Defense_Lawyer>, I represent the suspect in court.
US 09	As a <Defense_Lawyer>, I seek out all clients suspected of a crime to be not guilty.
US 10	As a <Defense_Lawyer>, I can appeal a losing case to the Court_Appeals, where I can get my client's case retried or thrown out for misconduct of the Prosecutor_Lawyer.
US 11	As a <Prosecutor_Lawyer>, I represent the state.
US 12	As a <Prosecutor_Lawyer>, I press charges onto the suspected person.
US 13	As a <Prosecutor_Lawyer>, I collect evidence from investigating officers.
US 14	As a <Prosecutor_Lawyer>, I provide evidence beyond a reasonable doubt to charge the suspect of a crime.
US 15	As a <Court>, I assign which cases go to Court_Superior, Court_District, Court_Appeal.
US 16	As a <Court>, I determine the date and time a case will appear in court.
US 17	As a <Court_Appeal>, I receive all cases given to me by the Court.
US 18	As a <Court_Appeal>, I hear both Offender_Felony status and Offender_Misdemeanor status after adjudication.
US 19	As a <Court_Appeal>, I hear only the evidence of Lawyer_Defense.
US 20	As a <Court_Appeal>, I determine if presented evidence is sufficient for a mis-trial.

US 21	As a <Court_Appeal>, I send cases to Court for a case retrial.
US 22	As a <Court_Superior>, I receive all cases given to me by the Court.
US 23	As a <Court_Superior>, I hear all cases in relation to a Offender_Felony status.
US 24	As a <Court_Superior>, I hear all cases which Lawyer_Prosecutor presents.
US 25	As a <Court_Superior>, I determine the legitimacy of evidence, by hearing the arguments of both Lawyer_Prosecutor and Lawyer_Defense.
US 26	As a <Court_Superior>, I determine case dismissal.
US 27	As a <Court_Superior>, I determine the sentencing of suspects (if they are found guilty).
US 28	As a <Court_District>, I receive all cases given to me by the Court.
US 29	As a <Court_District>, I hear all cases in relation to a Offender_Misdemeanor status.
US 30	As a <Court_District>, I hear all cases which Lawyer_Prosecutor presents.
US 31	As a <Court_District>, I determine the legitimacy of evidence, by hearing the arguments of both Lawyer_Prosecutor and Lawyer_Defense.
US 32	As a <Court_District>, I determine case dismissal.
US 33	As a <Court_District>, I determine the sentencing of suspect (if they are found guilty).
US 34	As an <Offender>, I can file for Court_Appeals.
US 35	As an <Offender>, I can have a Lawyer_Defense.
US 36	As an <Offender>, I have access to only my casefile.
US 37	As an <Offender_Felony>, I can file for Court_Appeals.
US 38	As an <Offender_Felony>, I can have a Lawyer_Defense.
US 39	As an <Offender_Felony>, I have access to only my casefile.
US 40	As an <Offender_Misdemeanor>, I can file for
US 41	As an <Offender_Misdemeanor>, I can have a Lawyer_Defense.
US 42	As an <Offender_Misdemeanor>, I have access to only my casefile.
US 43	As a <Citizen>, I can access public Court information.
US 44	As a <Citizen_Suspect>, I will have a Lawyer_Defendent.
US 45	As a <Citizen_Suspect>, I will have a case with the Court.

CONCEPTUAL DESIGN

Include your complete updated conceptual design here. Use the format shown below.

Entity: **CrimeCodes**
Attributes:

PK*Codes
Offense
OffenseType
SentenceRange

Entity: **CaseFiles**
Attributes:
PK*FileNumber
TrialType
AdjudicationStatus
FK*Codes

Entity: **Citizen**
Attributes:
PK* SSN
FirstName
LastName
Sentence
ChargeDate

Entity: **ConsolidatedCase**
Attributes:
PK* (SSN, FileNumber)
FK* SSN
FK* FileNumber

Entity: **DefenseLawyer**
PK*BarID
FirstName
LastName
PhoneNumber
Email
Fax
AgencyType
AgencyName

Entity: **LawyerAssig**
Attributes:

PK*(SSN, BarID)
FK*SSN
FK*BarID

Entity: **GovernmentEmployee**
Attributes:
PK*Employee_ID
FirstName
LastName
Position

Entity: **EmployeeAssig**
Attributes:
PK*(FileNumber, EmployeeID)
FK* FileNumber
FK* EmployeeID

Entity: **Court**
Attributes:
PK*CourtID
CourtType

Entity: **CourtSession**
Attributes:
PK*(courtDate, FileNumber, CourtID)
courtDate
FK* FileNumber
FK* CourtID

Entity: **JudgeAssig**
Attributes:
PK*(AssignDate, EmployeeID, CourtID)
AssignDate
FK* EmployeeIF
FK* CourtID

Relationship: Entity1 relationship-phrase Entity2
Relationship: **CaseFiles** -has- **CrimeCodes**

Cardinality: **Many to One**
Participation:
Casefiles has total participation
CrimeCodes has total participation

Relationship: **CaseFiles** -has a- **Citizen**
Cardinality: **Many** to **Many**
Participation:
CaseFiles has total partial - not all cases have a known suspect, therefore partial participation
ConsolidatedCase has total participation

Relationship: **CaseFiles** -takes place in- **Court**
Cardinality: **Many** to **Many**
Participation:
CaseFiles has partial - cases that do not have a citizen do not have a court, court session, or judge. Also known as cold cases.
Court has total participation

Relationship: **CaseFiles** -assigned to- **GovernmentEmployee**
Cardinality: **Many to Many**
Participation:
CaseFiles has partial - closed cases, or unsolvable cases will be unassigned and sent to the archives to be reassigned in the future
GovernmentEmployee has total participation

Relationship: **GovernmentEmployee** -has- **Court**
Cardinality: **Many to Many**
Participation:
GovernmentEmployee has total participation
Court has total participation

Relationship: **DefenseLawyers** -represents- **Citizens**
Cardinality: **One to Many**
Participation

DefenseLawyers has total participation
Citizens has partial - not all citizens will have a defense lawyer, because some people choose to self-represent

LOGICAL DESIGN WITH HIGHEST NORMAL FORMS AND INDEXES

Include your complete updated logical design here. Use the format shown below.

Table: **CrimeCodes**
Columns:
PK* Codes
Offense
OffenseType
SentenceRange
Highest normalization level: BCNF
Indexes: idx_Codes
Index: <clustered>
Columns: <Codes>
Justification: Codes is primary key and unique, and data is logical sorted on the key

Table: **CaseFiles**
Columns:
PK* FileNumber
TrialType
AdjudicationStatus
FK* Codes (Foreign key; references **pk Codes** of **CrimeCodes**)
Highest normalization level: BCNF
Indexes: idx_FileNumber, idx_Codes
Index: <clustered on FileNumber, non-clustered on Codes>
Columns: <FileNumber, Codes>
Justification: FileNumber is primary key and unique, and data is logical sorted on the key. Codes is used in joins and views.

Table: **Citizen**
Columns:
PK* SSN

FirstName
LastName
Sentence
ChargeDate

Highest normalization level: BCNF

Indexes: idx_SSN, idx_ChargeDate

Index: <clustered on SSN, ChargeDate non-clustered>

Columns: <SSN, Charge Date>

Justification: SSN is primary key and unique, and data is logical sorted on the key. ChargeDate would be used in timeline based queries.

Table: **ConsolidatedCase**

Columns:

PK* (SSN, FileNumber)

FK* SSN [Foreign key; References **pk SSN** of Citizens]

FK* FileNumber [Foreign key; References **pk FileNumber** of CaseFiles]

Justification: Junction table for many to many relationship of casefiles to citizens

Highest normalization level: BCNF

Indexes: idx_SSN_FileNumber, idx_SSN, idx_FileNumber

Index: <clustered on composite key, non-clustered on FileNumber and SSN>

Columns: <SSN, FileNumber>

Justification: (FileNumber, SSN) is the composite primary key and unique, and data is logical sorted on the key. Indexing both individually will improve joins and views as well.

Table: **DefenseLawyer**

Columns:

PK* BarID

FirstName

LastName

PhoneNumber

Email

Fax

AgencyType

AgencyName

Highest normalization level: BCNF

Indexes: idx_BarID, idx_AgencyType

Index: <clustered on BarID, non-clustered on AgencyType>

Columns: <BarID, AgencyType>

Justification: BarID is primary key and unique, and data is logical sorted on the key. AgencyType is used in view_PublicDefenseLawyer

Table: **LawyerAssig**

Columns:

PK* (SSN, BarID)

FK* SSN [Foreign key; References **pk SSN** of Citizens]

FK* BarID [Foreign key; References **pk BarID** of DefenseLawyer]

Justification: Junction table for one to many relationship of Citizen and DefenseLawyer

Highest normalization level: BCNF

Indexes: idx__SSN_BarID, idx_BarID

Index: <clustered on pk, non-clustered on BarID>

Columns: <SSN, BarID>

Justification: SSN, and EmployeeID is the composite primary keys and unique, and data is logical sorted on the key. BarID is used in the procedure get_RepresentedCitizens.

Table: **GovernmentEmployee**

Columns:

PK* EmployeeID

FirstName

LastName

Position

Highest normalization level: BCNF

Indexes: idx_EmployeeID, idx_Posistion

Index: <clustered on EmployeeID, non-clustered on Position>

Columns: <BarID, AgencyType>

Justification: EmployeeID is primary key and unique, and data is logical sorted on the key. Position is used in multiple views.

Table: **EmployeeAssig**

Columns:

PK* (FileNumber, EmployeeID)
FK* FileNumber [Foreign key; References **pk FileNumber** of CaseFiles]
FK* EmployeeID [Foreign key; References **pk EmployeeID** of GovernmentEmployee]

Justification: Junction table for one to many relationship of Casefiles and GovernmentEmployee

Highest normalization level: BCNF

Indexes: idx__FileNumber_EmployeeID, idx_EmployeeID

Index: <clustered on pk, non-clustered on EmployeeID>

Columns: <FileNumber, EmployeeID>

Justification: FileNumber, and EmployeeID is the composite primary keys and unique, and data is logical sorted on the key. EmployeeID is used in procedures so indexing it individually will help speed.

Table: **Court**

Columns:

PK* CourtID
CourtType

Highest normalization level: BCNF

Indexes: idx__CourtID

Index: <clustered>

Columns: <CourtID>

Justification: CourtID is primary key and unique, and data is logical sorted on the key.

Table: **CourtSession**

Columns:

PK*(CourtDate, FileNumber, CourtID)
FK*FileNumber [Foreign key; References **pk FileNumber** of CaseFiles]
FK* CourtID [Foreign key; References **pk** CourtID of Court]
CourtDate

Justification: Many cases will be held in a single day, and one case may be seen on several court dates and in several Courts

Highest normalization level: BCNF

Indexes: idx__CourtDate_FileNumber_CourtID, idx_FileNumber

Index: <clustered on pk, non-clustered on FileNumber>

Columns: <CourtDate,FileNumber, EmployeeID>

Justification: FileNumber, and EmployeeID is the composite primary keys and unique, and data is logical sorted on the key. Adding a separate index on FileNumber will improve lookup speed.

Table: **JudgeAssig**

Columns:

PK*(AssignedDate, EmployeeID, CourtID)

FK*EmployeeID[Foreign key; References **pk EmployeeID** of GovernmentEmployee]

FK*CourtID[Foreign key; References **pk CourtID** of Court]

AssignedDate

Justification: Judges often rotate daily in which court they will be attending, therefore needing a combination of the Date and EmployeeID to allow for uniqueness of each row.

Highest normalization level: BCNF

Indexes: idx__AssignedDate_EmployeeID_CourtID, idx_EmployeeID

Index: <clustered on pk, non-clustered on EmployeeID>

Columns: <AssignedDate, EmployeeID, CourtID >

Justification: AssignedDate, EmployeeID, and CourtID is the composite primary keys and unique, and data is logical sorted on the key. A separate index on EmployeeID will improve performance since this table will be commonly filtered or joined on EmployeeID

VIEWS AND STORED PROGRAMS

List the views relevant to your application here. Use the format specified below.

View: <view_judges>

```
CREATE VIEW view_judges
AS
SELECT FirstName, LastName
FROM GovernmentEmployee
WHERE Position = 'Judge';
```

Goal: <As a government employee, I want to view the names of all the judges>

View: <view_citizens>

```
CREATE VIEW view_citizens
AS
SELECT FirstName, LastName
FROM Citizen;
```

Goal: <As a government employee, I want to view the names of the citizens>

View: <view_felons>

```
CREATE VIEW view_felons
AS
SELECT FirstName, LastName
FROM Citizen
WHERE SSN IN (
        SELECT SSN
        FROM ConsolidatedCase
        WHERE FileNumber IN (
                SELECT FileNumber
                FROM CaseFiles
                WHERE Codes IN (
                        SELECT Codes
                        FROM CrimeCodes
                        WHERE OffenseType = 'F'

                )
        )
);
```

Goal: <As a government employee, I want to see the list of all felons>

VIEW: <view_prosecutor>

```
CREATE VIEW view_prosecutor
AS
SELECT FirstName, LastName
FROM GovernmentEmployee
WHERE Position = 'Prosecutor';
```

Goal: <As a government employee, I want to see all potential prosecutors for a trial>

VIEW: <view_PublicDefenseLawyer>

```
CREATE VIEW view_PublicDefenseLawyer
AS
```

```
SELECT FirstName, LastName, PhoneNumber, Email
FROM DefenseLawyer
WHERE AgencyType = 'Public';
```

Goal: <As a suspected citizen, I want to see and contact potential representation>

VIEW: <view_felonies>

```
CREATE VIEW view_felonies
AS
SELECT Code, Offense, SentenceRange
FROM CrimeCodes
WHERE OffenseType = 'F';
```

```
CREATE VIEW view_felons AS
SELECT
    Citizen.FirstName, Citizen.LastName, CrimeCodes.offense
FROM
    Citizen
INNER JOIN ConsolidatedCase ON Citizen.SSN = ConsolidatedCase.SSN
INNER JOIN CaseFiles ON ConsolidatedCase.FileNumber =
CaseFiles.FileNumber
INNER JOIN CrimeCodes ON CaseFiles.Codes = CrimeCodes.Codes
WHERE
    CrimeCodes.OffenseType = 'F';
```

Goal: <As a member of the public, , I want to see information on the types of felonies>

VIEW: <view_misdemeanors>

```
CREATE VIEW view_misdemeanors
AS
SELECT Code, Offense, SentenceRange
FROM CrimeCodes
WHERE OffenseType = 'M';
```

Goal: <As a member of the public, I want to see information the types of misdemeanors>

VIEW: <view_Robberies>
CREATE VIEW view_Robberies
AS
SELECT *
FROM CaseFiles
WHERE Codes IN (1.0, 1.1, 1.2, 1.3);

Goal: <As a crime analyst, I want to see the details of committed robberies>

VIEW: <view_PendingCases>
CREATE VIEW view_PendingCases
AS
SELECT *
FROM CaseFiles
WHERE AdjucationStatus = False;

Goal: <As part of the court, I want to see the status of all pending cases to know which ones to process and which ones to schedule>

VIEW: <view_CaseOffenses>
CREATE VIEW view_CaseOffenses
AS
SELECT CaseFile.FileNumber, CaseFile.TrialType, CrimeCodes.Offense,

CrimeCodes.OffenseType
INNER JOIN CrimeCodes ON CaseFile.Codes = CrimedCodes.Codes;

Goal: <As a government employee, I want to view the file numbers of cases, with their trial type as well as offense and offense type>

Stored Procedure: <get_FileNumber>
Parameters: <SSN>
Goal: <As a government employee, I want to see a citizen's file number>

```
DELIMITER//

CREATE PROCEDURE get_FileNumber (IN SSN var(50))
BEGIN
SELECT FileNumber
FROM ConsolidatedCase
WHERE SSN = SSN;
END//

DELIMITER;
```

Stored Procedure: <get_RepresntedCitizens>
Parameters: <para_BardID>
Goal: <As a lawyer, I want to see the list of all the citizens I represent>

```
DELIMITER//
CREATE PROCEDURE get_RepresentedCitizens (IN para_BarID var(50))
BEGIN
SELECT FirstName, LastName
FROM Citizens
WHERE SSN IN (
        SELECT SSN
        FROM LawyerAssig
        WHERE BarID = para_BarID;
)
END//
DELIMITER;
```

Stored Procedure: <get_AssignedEmployees>
Parameters: <para_FileNumber>
Goal: <As a government employee, I want so see which employees worked on a specific case>

```
DELIMITER//
CREATE PROCEDURE get_AssignedEmployees (IN para_FileNumber var(50))
BEGIN
SELECT EmployeeID, FirstName, LastName, Position
```

```
FROM GovernmentEmployee
WHERE EmployeeID IN (
SELECT EmployeeID
        FROM EmployeeAssig
        WHERE FileNumber = para_FileNumber
);
END//
DELIMITER;
```

Stored Procedure: <get_Judges>
Parameters: <para_FileNumber>
Goal: <As a government employee I want to see the judges, that worked on a specific case>

```
DELIMITER//
CREATE PROCEDURE get_Judges (IN para_FileNumber var(50))
BEGIN
SELECT FirstName, LastName
FROM GovernmentEmployee
WHERE Position = 'Judge' AND EmployeeID IN (
SELECT EmployeeID
        FROM EmployeeAssig
        WHERE FileNumber = para_FileNumber
);
END//
DELIMITER;
```

Stored Function: <get_NumberOfCases>
Parameters: <para_EmployeeID>
Goal: Count how many cases a singular Employee has

```
DELIMITER//
CREATE FUNCTION get_NumberOfCases(para_EmployeeID varchar(50))
RETURNS INT
DETERMINISTIC
BEGIN
    DECLARE case_count INT;
    SELECT COUNT(*)
    INTO case_count
```

```
    FROM EmployeeAssig
    WHERE EmployeeID = para_EmployeeID;
    RETURN case_count;
END//
DELIMITER;
```

Goal: <As a government employee, I want to find the number of cases an employee has worked on>

Stored Function: <get_NumberOfEmployees>
Parameters: <para_Position>
Goal:Goal: <As a government employee, I want to find out how many employees of each position there are>

```
DELIMITER//
CREATE FUNCTION get_NumberOfEmployees(para_Position varchar(50))
RETURNS INT
DETERMINISTIC
BEGIN
    DECLARE emp_count INT;
    SELECT COUNT(*)
    INTO emp_count
    FROM GovernmentEmployee
    WHERE Position = para_Position;
    RETURN emp_count;
END//
DELIMITER;
```

Stored Trigger: <drop_CaseFiles> on CaseFiles table and ConsolidatedCase Table
Goal: If a CaseFile is deleted, it will also be deleted in consolidated cases and in EmployeeAssig.

QLQuery1.sql - DE...LLQ\Jayanthi (68))*

```sql
CREATE VIEW view_felons
AS
SELECT FirstName, LastName
FROM Citizen
WHERE SSN IN (
SELECT SSN
FROM ConsolidatedCase
WHERE FileNumber IN (
    SELECT FileNumber
    FROM CaseFiles
    WHERE Codes IN (
        SELECT Codes
        FROM CrimeCodes
        WHERE OffenseType = 'F'

        )
    )
);
```

SQLQuery1.sql - DE...LLQ\Jayanthi (68))*

```
SELECT * FROM view_felons;
```

100 %

Results | Messages

	FirstName	LastName
1	Alex	Taylor
2	Isaac	Garcia
3	Haley	Bennett
4	Dylan	Ortiz

```sql
CREATE VIEW view_judges
AS
SELECT FirstName, LastName
FROM GovernmentEmployee
WHERE Position = 'Judge';
```

```sql
SELECT * FROM view_judges;
```

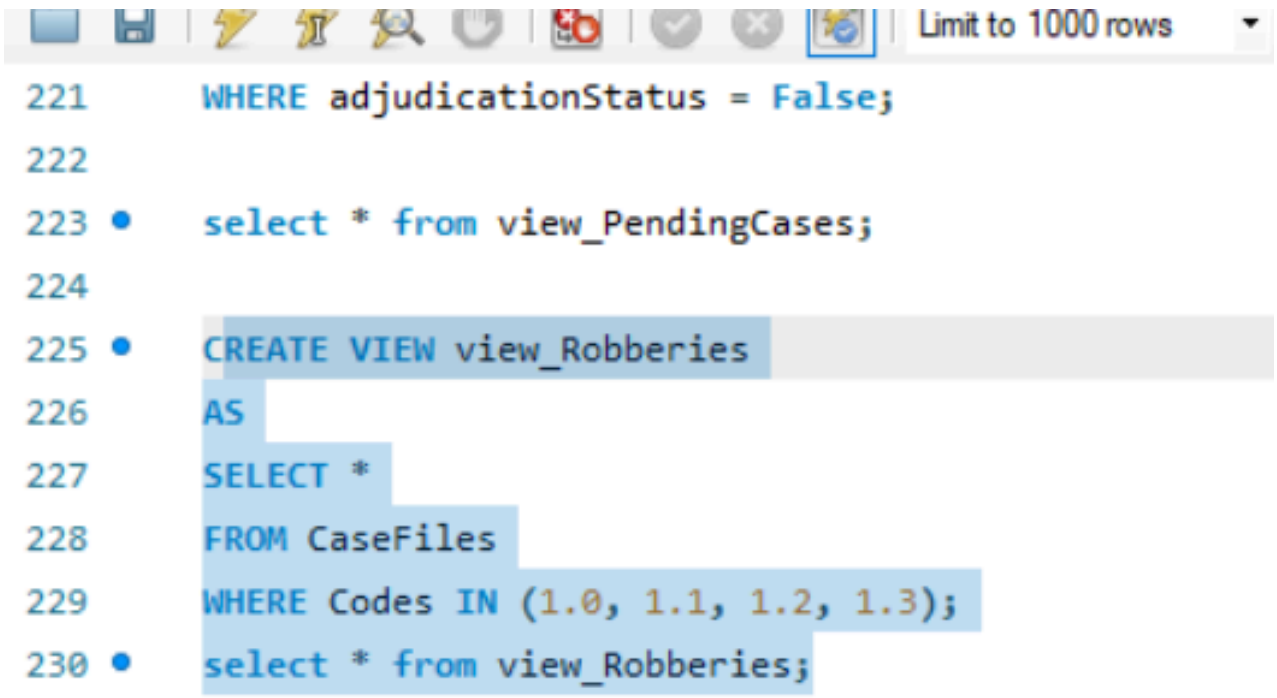
0 %

Results | Messages

	FirstName	LastName
	Sophia	Martinez
	Henry	Walker
	Amelia	Carter

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

fileNumber	trailType	adjudicationStatus	codes
2004CRS0002	Jury	0	1.2
2004CRS0010	Plea Bargain	0	2.0
2025CRS0004	Bench	0	1.3
2025CRS0005	Plea Bargain	0	2.1



Result Grid | Filter Rows: | Export: | Wrap Cell Co

fileNumber	trailType	adjudicationStatus	codes
2004CRS0006	Plea Bargain	1	1.0
2004CRS0002	Jury	0	1.2
2004CRS0009	Jury	1	1.2
2025CRS0004	Bench	0	1.3