Mid Term Project

Database Systems

Name:	Ahmad Zarrar
Roll no. :	(FA23-BDS-002)
Name:	Ibrahim Shoaib
Roll no. :	(FA23-BDS-018)

Instructor: Mr. Muhammad Haris

Comsats University

Islamabad

Crime Records Management System

This database system is designed to manage and maintain records related to criminal activities, individuals involved in crimes (criminals and witnesses), police officers responsible for handling cases, and detailed reports of each crime. The database aims to support law enforcement agencies in organizing, retrieving, and analyzing crime-related data efficiently.

Modules and Their Purpose

1. **Crimes Module** This module maintains information about every crime incident, including the type, severity, location, and date of occurrence. It acts as the central entity around which other modules revolve.

2. **Criminals Module** This module stores data about individuals identified as criminals. It includes personal information and a brief history of their criminal background.

3. **Police Officers Module** This module holds information about law enforcement officers. Each record includes personal details, rank, station details, and contact information.

4. **Crime Records Module** This module links crimes with criminals and the officers handling them. It includes the date of the crime, the conviction status, and sentencing information.

5. **Witnesses Module** This module stores personal details of witnesses who have provided statements or are linked to crimes.

6. **Witness-Crime Relationship Module** This module connects witnesses with the crimes they have witnessed. It helps track which individuals have testified or been associated with specific incidents.

Database Schema

1. **CRIMES**
    - CRIMEID: Unique identifier for each crime.
    - CRIMETYPE: Type or category of crime.
    - DATEOCCURRED: Date on which the crime took place.
    - LOCATION: Place where the crime occurred.
    - SEVERITY: Level of severity (e.g., low, medium, high).

2. **CRIMINALS**
    - CRIMINALID: Unique identifier for each criminal.
    - FIRSTNAME: First name of the criminal.
    - LASTNAME: Last name of the criminal.
    - DATEOFBIRTH: Birthdate of the criminal.
    - GENDER: Gender of the individual.
    - ADDRESS: Residential address.
    - CRIMEHISTORY: Summary of past crimes committed by the individual.

3. **POLICEOFFICERS**
    - OFFICERID: Unique identifier for each police officer.
    - FIRSTNAME: First name of the officer.
    - LASTNAME: Last name of the officer.
    - RANK: Rank held by the officer.
    - STATION: Name of the station they are affiliated with.
    - CONTACTINFO: Contact details.

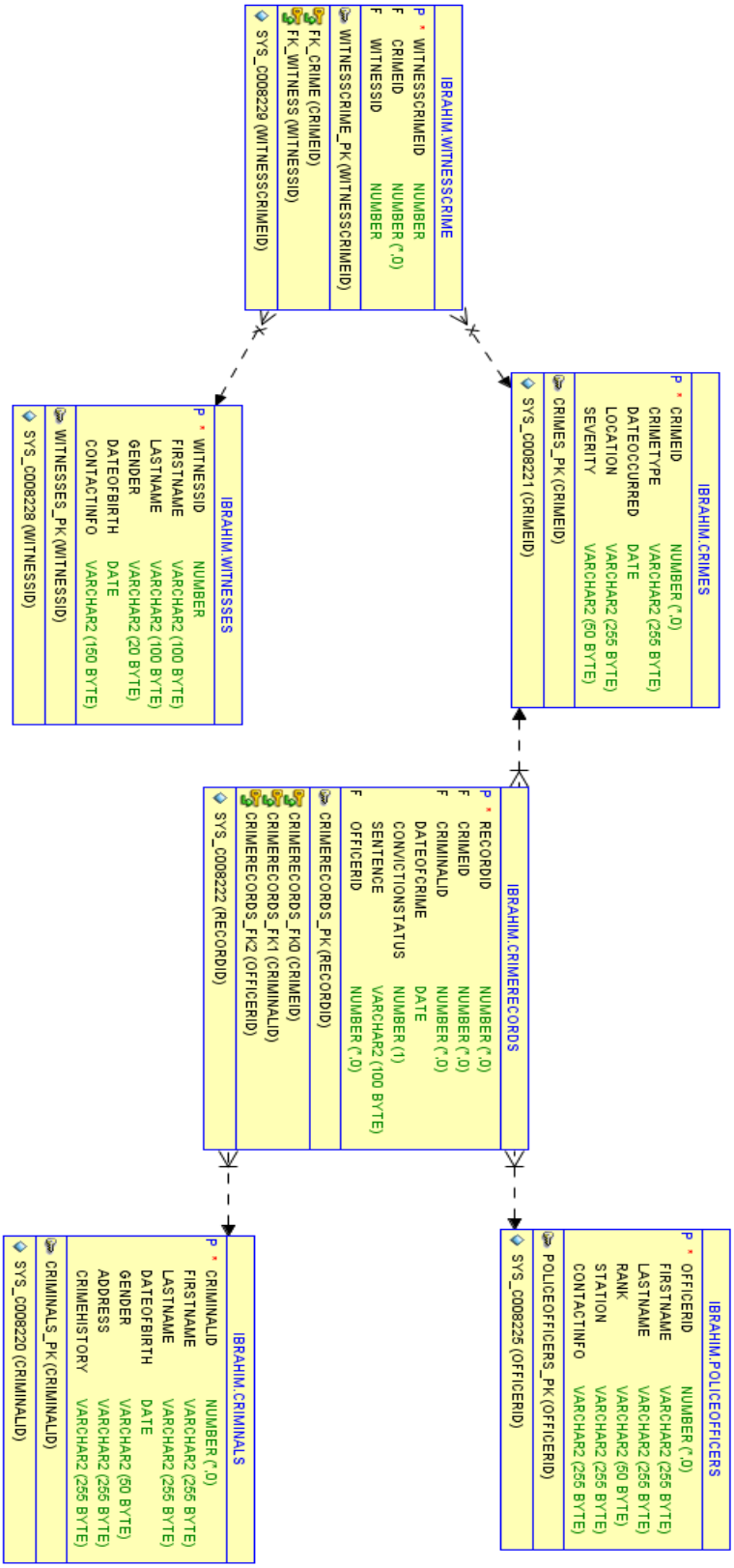
4. **CRIMERECORDS**
    - RECORDID: Unique identifier for each record.
    - CRIMEID: Reference to the crime associated with this record.
    - CRIMINALID: Reference to the criminal involved.
    - DATEOFCRIME: Date when the crime was committed.
    - CONVICTIONSTATUS: Status indicating whether the criminal was convicted (e.g., 0 = Not Convicted, 1 = Convicted).
    - SENTENCE: Details of the sentence awarded.
    - OFFICERID: Reference to the officer who handled the case.

5. **WITNESSES**
    - WITNESSID: Unique identifier for each witness.

- FIRSTNAME: First name of the witness.
- LASTNAME: Last name of the witness.
- GENDER: Gender of the witness.
- DATEOFBIRTH: Birthdate of the witness.
- CONTACTINFO: Contact details.

6. **WITNESSCRIME**
   - WITNESSCRIMEID: Unique identifier for each witness-crime relationship.
   - CRIMEID: Reference to the crime witnessed.
   - WITNESSID: Reference to the witness.

ERD


IBRAHIM.WITNESSCRIME
P * WITNESSCRIMEID NUMBER
F CRIMEID NUMBER (*,0)
F WITNESSID NUMBER
WITNESSCRIME_PK (WITNESSCRIMEID)
FK_CRIME (CRIMEID)
FK_WITNESS (WITNESSID)
SYS_C008229 (WITNESSCRIMEID)
IBRAHIM.CRIMES
P * CRIMEID NUMBER (*,0)
CRIMETYPE VARCHAR2 (255 BYTE)
DATEOCCURRED DATE
LOCATION VARCHAR2 (255 BYTE)
SEVERITY VARCHAR2 (50 BYTE)
CRIMES_PK (CRIMEID)
SYS_C008221 (CRIMEID)
IBRAHIM.WITNESSES
P * WITNESSID NUMBER
FIRSTNAME VARCHAR2 (100 BYTE)
LASTNAME VARCHAR2 (100 BYTE)
GENDER VARCHAR2 (20 BYTE)
DATEOFBIRTH DATE
CONTACTINFO VARCHAR2 (150 BYTE)
WITNESSES_PK (WITNESSID)
SYS_C008228 (WITNESSID)
IBRAHIM.CRIMERECORDS
P * RECORDID NUMBER (*,0)
F CRIMEID NUMBER (*,0)
F CRIMINALID NUMBER (*,0)
DATEOFCRIME DATE
CONVICTIONSTATUS NUMBER (1)
SENTENCE VARCHAR2 (100 BYTE)
F OFFICERID NUMBER (*,0)
CRIMERECORDS_PK (RECORDID)
CRIMERECORDS_FK0 (CRIMEID)
CRIMERECORDS_FK1 (CRIMINALID)
CRIMERECORDS_FK2 (OFFICERID)
SYS_C008222 (RECORDID)
IBRAHIM.POLICEOFFICERS
P * OFFICERID NUMBER (*,0)
FIRSTNAME VARCHAR2 (255 BYTE)
LASTNAME VARCHAR2 (255 BYTE)
RANK VARCHAR2 (50 BYTE)
STATION VARCHAR2 (255 BYTE)
CONTACTINFO VARCHAR2 (255 BYTE)
POLICEOFFICERS_PK (OFFICERID)
SYS_C008225 (OFFICERID)
IBRAHIM.CRIMINALS
P * CRIMINALID NUMBER (*,0)
FIRSTNAME VARCHAR2 (255 BYTE)
LASTNAME VARCHAR2 (255 BYTE)
DATEOFBIRTH DATE
GENDER VARCHAR2 (50 BYTE)
ADDRESS VARCHAR2 (255 BYTE)
CRIMEHISTORY VARCHAR2 (255 BYTE)
CRIMINALS_PK (CRIMINALID)
SYS_C008220 (CRIMINALID)

Basic Operations of the Crime Database System

1. Add (Insert) Operation

```
insert into criminals(CRIMINALID,

FIRSTNAME,

LASTNAME,

DATEOFBIRTH,

GENDER,

ADDRESS,

CRIMEHISTORY)


values (21, 'maxine', 'arnold', null, 'female', null, null);
```

2. View (Select) Operation

```
SELECT C.FIRSTNAME, C.LASTNAME, R.SENTENCE

FROM CRIMINALS C;
```

3. Update Operation

```
UPDATE POLICEOFFICERS

SET CONTACTINFO = '0300-9876543'

WHERE OFFICERID = 5;
```

4. Delete Operation

```
DELETE FROM WITNESSES

WHERE WITNESSID = 8;
```

Queries

1. Joining 3 Tables with a Subquery

English: I need to see the names of all crimes seen by witness ID 1.

```
SELECT crimeid, crimetype, dateoccurred, location, severity
FROM crimes
NATURAL JOIN witnesscrime
NATURAL JOIN (
    SELECT *
    FROM witnesses
    WHERE witnessid = 1
);
```

Explanation: You're joining `crimes`, `witnesscrime`, and a subquery of `witnesses` filtered to only include witness ID 1. This gives you crime info seen by that witness.

---

2. Join with Grouping and Aggregation

English: I need to see the number of officers that have dealt with each crime.

```
SELECT crimetype, COUNT(officerid) AS officer_count
FROM crimes
NATURAL JOIN crimerecords
NATURAL JOIN policeofficers
GROUP BY crimetype;
```

Explanation: After joining related tables, this groups by `crimetype` and counts how many officers are associated with each crime.

---

3. Using sub query

English: Show the record IDs and crime types of crimes that had more than one witness.

```
SELECT recordid, crimetype
FROM crimerecords
NATURAL JOIN crimes
NATURAL JOIN (
    SELECT crimeid
    FROM (
        SELECT crimeid, COUNT(witnesscrimeid) AS witnesscount
        FROM witnesscrime
```

```
        GROUP BY crimeid
    )
    WHERE witnesscount > 1
);
```

Explanation: A nested subquery first counts witnesses per crime, then filters those with more than 1. The outer query shows the full crime record for those.

---

4. Using set operator

English: Select all criminals who have also been a witness.

```
SELECT criminalid, firstname
FROM criminals

INTERSECT

SELECT witnessid, firstname
FROM witnesses;
```

Explanation: Using `INTERSECT` here returns only those people who exist in both `criminals` and `witnesses`.

Extra Queries

1) Highest, Lowest, Sum, and Average Number of Crimes Handled by Officers

```
SELECT
    MAX(COUNTED) AS Maximum,
    MIN(COUNTED) AS Minimum,
    SUM(COUNTED) AS Sum,
    ROUND(AVG(COUNTED)) AS Average
FROM (
    SELECT
        OFFICERID,
        COUNT(*) AS COUNTED
    FROM
        CRIMERECORDS
    GROUP BY
        OFFICERID
);
```

---

2) Difference Between Maximum and Minimum Crimes Committed by a Criminal

```
SELECT
    (MAX(CRIME_COUNT) - MIN(CRIME_COUNT)) AS DIFFERENCE
FROM (
    SELECT
        CRIMINALID,
        COUNT(*) AS CRIME_COUNT
    FROM
        CRIMERECORDS
    GROUP BY
        CRIMINALID
);
```

---

3) Officer and Criminal Involved in the Least Crimes (Filtered for More Than 2 Crimes)

```
SELECT
    OFFICERID,
    MIN(CRIME_COUNT) AS Crime_Count
FROM (
    SELECT
        OFFICERID,
        CRIMINALID,
        COUNT(*) AS CRIME_COUNT
    FROM
        CRIMERECORDS
    GROUP BY
        OFFICERID, CRIMINALID
)
WHERE
    CRIME_COUNT > 2
GROUP BY
    OFFICERID
ORDER BY
    Crime_Count DESC;
```

---

4) Show All Police Officers with Their Station

```
SELECT
    LASTNAME,
    STATION
FROM
    POLICEOFFICERS;
```

---

5) Count the Number of Crimes per Location

```
SELECT
    LOCATION,
    COUNT(*) AS TotalCrimes
FROM
    CRIMES
GROUP BY
    LOCATION
ORDER BY
    TotalCrimes DESC;
```

---

6) All Convicted Criminals Along with Their Sentence and Crime Type

```
SELECT
    FIRSTNAME,
    LASTNAME,
    CRIMETYPE,
    SENTENCE
FROM
    CRIMERECORDS
JOIN
    CRIMINALS ON CRIMERECORDS.CRIMINALID = CRIMINALS.CRIMINALID
JOIN
    CRIMES ON CRIMERECORDS.CRIMEID = CRIMES.CRIMEID
WHERE
    CRIMERECORDS.CONVICTIONSTATUS = 1;
```