Case Study and Database Design for Police Forces of England and Whales

Student Name1: Aashutosh Thapa

Student id: c7466915

Student Name2: Ojaswi Shrestha

Student id: c7466889

Content Page

Introduction to the Report…………………………………………………………..3

Task 1: Conceptual Design………………………………………………………...4

1.2 Extended Entity Relationship Diagram……………………………………….4

1.3 Definition for each entity………………………………………………………..5

1.4 Occurrences for each entity……………………………………………………9

Task 2: Logical Design…………………………………………………………….11

Task 3: Physical Design…………………………………………………………..19

Task 4: Table Implementation…………………………………………………….20

Bibliography………………………………………………………………………...34

Introduction to the Report

In today's modern world where multiple businesses and organization depend upon the data storage and its management in order to perform operations in efficient manner. The purpose of this project is to store the criminal records, people and all the actions revolving around that related crime.

The database was designed using Entity Relationship Model, going through the process of physical and logical design and finally executed with SQL. The key usage of this database is to provide information on recorded crimes in easy and understandable way that even a regular person would be able to comprehend it through the use of proper interface.

Task 1: Conceptual Design

Crime and related entities:

* Each crime “Crime” occurs in certain location “Location” which is involves a unique identity “LocationID” and “CrimeID”.
* A crime can be associated with multiple object “Object”, person “Person”, and different events “Event”.

Detective and police officers:

* Police officers “PoliceOfficer” are associated with one or multiple crimes which they report to their supervisor i.e. inspectors “Inspector”.
* A detective is assigned to single or multiple crime which they are resopnsivle to solve

Offenders and history

* A person “Person” can be involved to multiple event through “Person\_Event” relationships.
* Each Offender is giver unique “OffenderID” which can be helpful in identifying the crime they have committed through the “OffenderHistory” entity.

1.2 Extended Entity Relationship Diagram (EER Diagram)

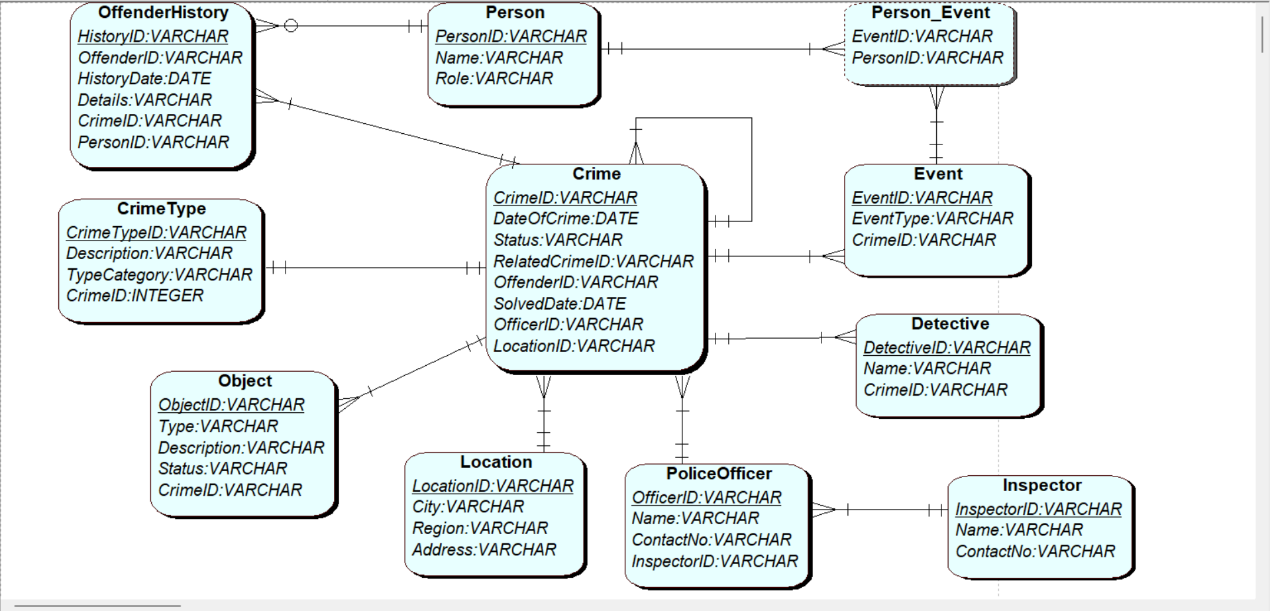
****

Fig: EERD of the Crime Database

1.3 Definition for each entity of the Crime database

* **Crime**

Description: This crime entity holds the major information about each crime that had been committed such as its date as well as its status of whether its ongoing or closed.

Attributes: CrimeID, DateOfCrime, Status (Open/Closed), RelatedCrimeID (if related to another crime), OffenderID (if solved), SolvedDate(If solved), OfficerID, LocationID

Primary key: CrimeID

Foreign key: LocationID, RelatedCrimeID, OffenderID, OfficerID

Relationship:

* + - Many-to-One: Crime → Location
    - Many-to-One: Crime → PoliceOfficer
    - One-to-One: Crime → CrimeType
    - One-to-Many: Crime → Event
    - One-to-Many: Crime → Object
    - One-to-One: Crime → Detective
    - One-to-Many: Crime → OffenderHistory

.

* **Location**

Description: The location entity holds Information about geography where the crime had occurred .

Attributes: LocationID, City, Region, Address

Primary key: LocationID

Foreign key: null (no foreign key)

Relationship:

* One-to-Many: Location → Crime

* **PoliceOfficer**

Description: The PoliceOfficer entity holds the details about the police.

Attributes: OfficerID, Name, ContactNo, InspectorID (Inspector they report to)

Primary key: OfficerID,

Foreign key: InspectorID

Relationship:

* Many-to-One: PoliceOfficer → Inspector
* Many-to-One: Crime → PoliceOfficer (Officers who record the crime)
* **Person**

Description: The Person entity holds the information about the person involved in the crime whether it be witness, victim, as well as offender.

Attributes: PersonID, Name, Role (Witness, Victim, Offender), ContactInfo, Address

Primary key: PersonID

Foreign key: null (no foreign key)

Relationship:

* Many-to-Many: Event → Person (Involvement in events like witnessing or reporting)

{On resolving many to many introduced new table name”Person\_Event”

So, the new relation is:

One-to-Many: Person → Person\_Event

}

* Many-to-One(optional at many): OffenderHistory → Person (If the person is an offender)
* **Event**

Description: The Event entity holds the information of specific incident related to the crime including the witness report.

Attributes: EventID, CrimeID, Description, EventType (Witness Report, Victim Report, etc.)

Primary key: EventID

Foreign key: CrimeID

Relationship:

* + - Many-to-One: Event → Crime
    - Many-to-Many: Event → Person

{On resolving many to many introduced new table name”Person\_Event”

So, the new relation is:

One-to-Many: Event → Person\_Event

* **Object**

Description: The Object Entity holds Information objects related to the crime such as weapons or evidences

Attributes: ObjectID , Type (evidence, weapon, etc.), Description, Status (Recovered, Missing, etc.), CrimeID

Primary key: ObjectID

Foreign key: CrimeID

Relationship:

* Many-to-One: Object → Crime
* **CrimeType**

Description: the CrimeType entity specifies the classification of crime typically between primary or secondary type

Attributes: CrimeTypeID, Description, TypeCategory (Primary, Secondary),CrimeID

Primary key: CrimeTypeID

Foreign key: CrimeID

Relationship:

* One-to-One: Crime → CrimeType
* **OffenderHistory**

Description: This OffenderHistory entity records the history of crimes committed by the offenders, including the crime the offenders have been part of with its detail

Attributes: HistoryID, OffenderID, HistoryDate, Details, CrimeID, PersonID

Primary key: HistoryID

Foreign key: CrimeID, PersonID

Relationship:S

* Many-to-One(optionality at many): OffenderHistory → Person (The offender)
* Many-to-One: OffenderHistory → Crime (The related crime)
* **Detective**

Description: This Detective entnty represents detectives who are assigned to investigate certain crime.

Attributes: DetectiveID, Name, CrimeID

Primary key: DetectiveID

Foreign key : CrimeID

Many-to-Many: Crime → CrimeType

Relationship:

* Many-to-One: Crime → Detective (Investigating detective or detective who solved the case)
* **Inspector**

Description: The location entity holds the details about the inspector

Attributes: InspectorID, Name, ContactNo

Primary key: InspectorID

Foreign key: null (no foreign key)

Relationship:

* One-to-Many: Inspector → PoliceOfficer
* **Person\_Event**

Description: The Person\_Event entity is created to resolve the many to many relationship of the entity “Person” and “Event”.

Attribute:EventID,PersonID

Primary key: **(**EventID,PersonID)

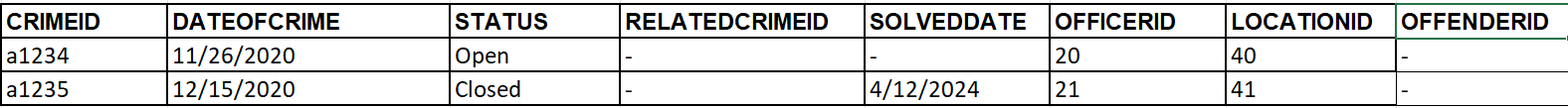
Foreign key:EventID,PersonID

Relationship:

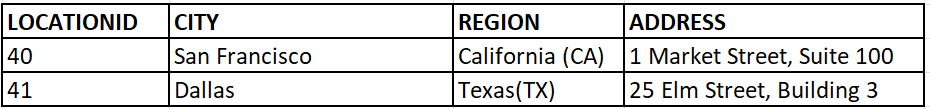
* Many-to-One: Person\_Event → Event
* Many-to-One: Person\_Event → Person

1.4 Occurrences for each entity of the Crime Database

**Crime**

****

**Location**

****

**PoliceOfficer**

**A white rectangular box with black text

Description automatically generated**

**Person**

**A close up of a sign

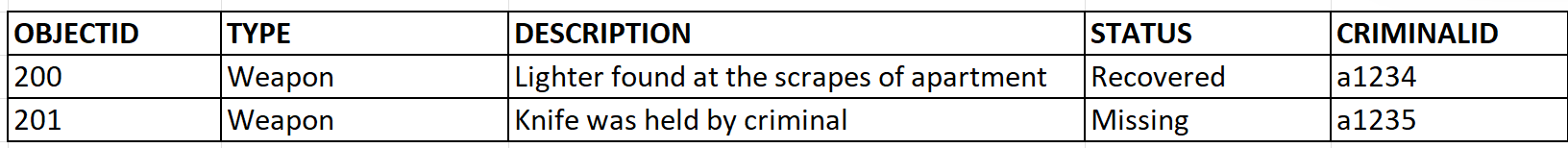
Description automatically generated**

**Event**

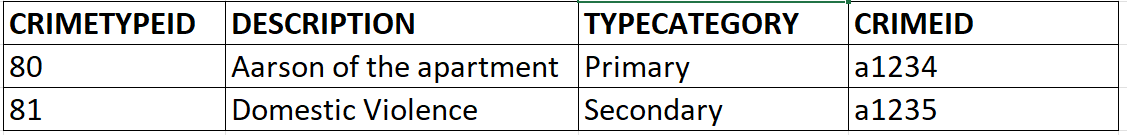
**A close up of a sign

Description automatically generated**

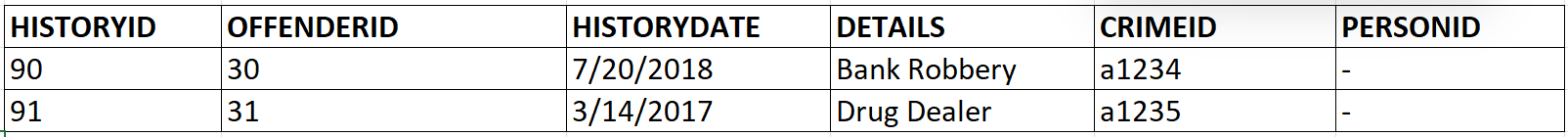
**Object**

****

**CrimeType**

****

**OffenderHistory**

****

**Detective**

**A screenshot of a computer

Description automatically generated**

**Inspector**

**A black and white text

Description automatically generated with medium confidence**

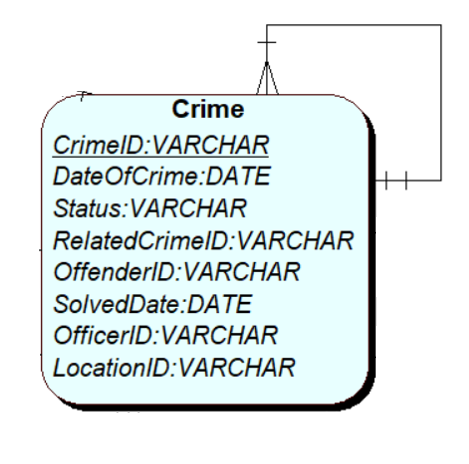
**Person\_Event**

**A white paper with black text

Description automatically generated**

**Task 2: Logical Design**

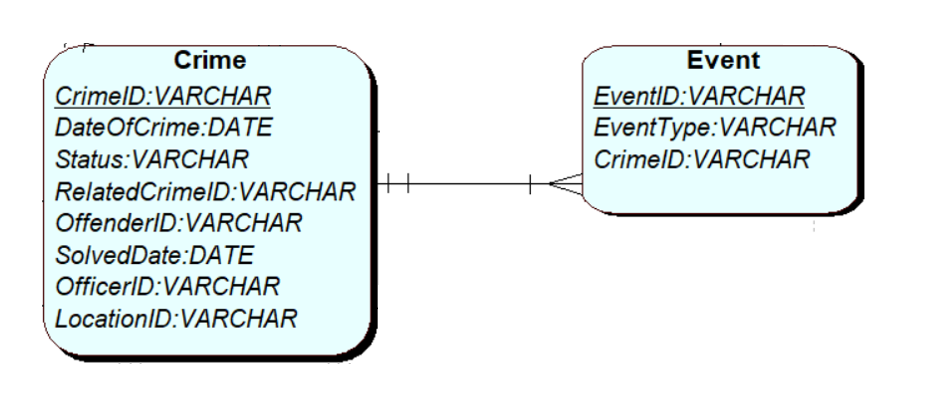
* Recursive relation of Crime entity with itself in case of a past crime being related to the present one.

****

Crime (CrimeID, DateOfCrime, Status, RelatedCrimeID, OffenderID, SolvedDate, OfficerID, LocationID)

Crime (CrimeID, DateOfCrime, Status, *RelatedCrimeID,* OffenderID, SolvedDate, OfficerID, LocationID)

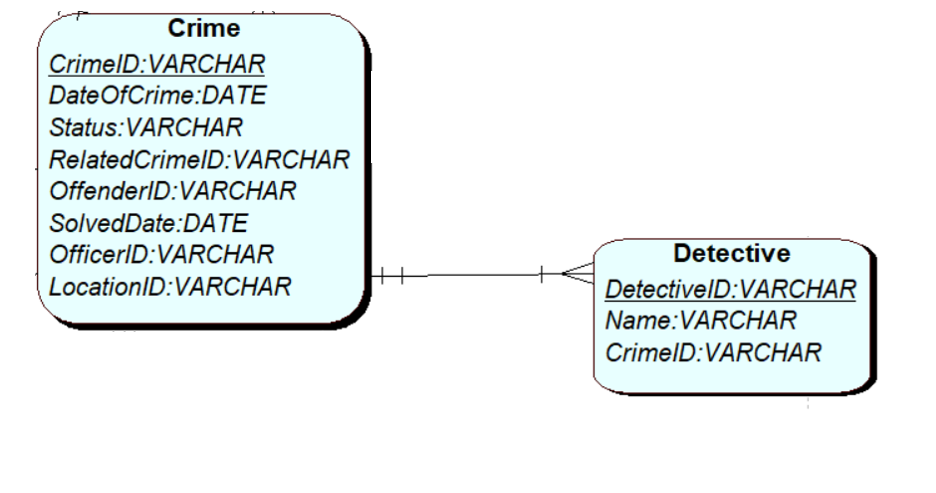
* One to many relation of Crime to Event entity where foreign key lies at Event entity

****

Crime (CrimeID, DateOfCrime, Status, RelatedCrimeID, OffenderID, SolvedDate, OfficerID, LocationID)

Event (EventID, EventType, *CrimeID)*

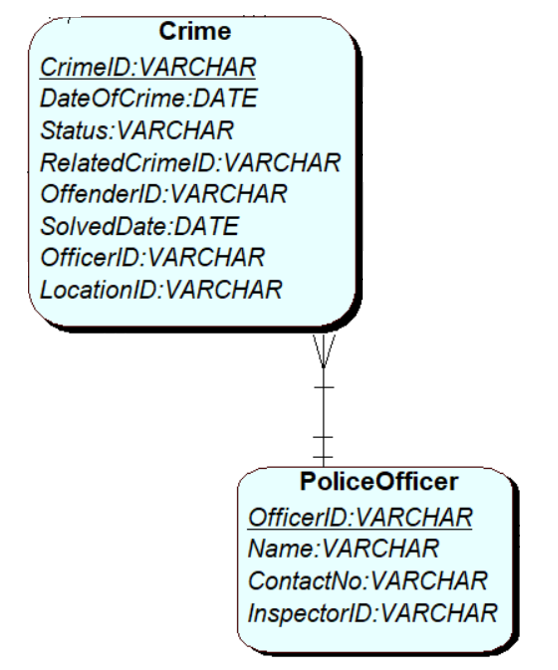
* One to many relation of Crime to Detective entity where foreign key lies at Detective entity



Crime (CrimeID, DateOfCrime, Status, RelatedCrimeID, OffenderID, SolvedDate, OfficerID, LocationID)

Detective (DetectiveID, Name, *CrimeID)*

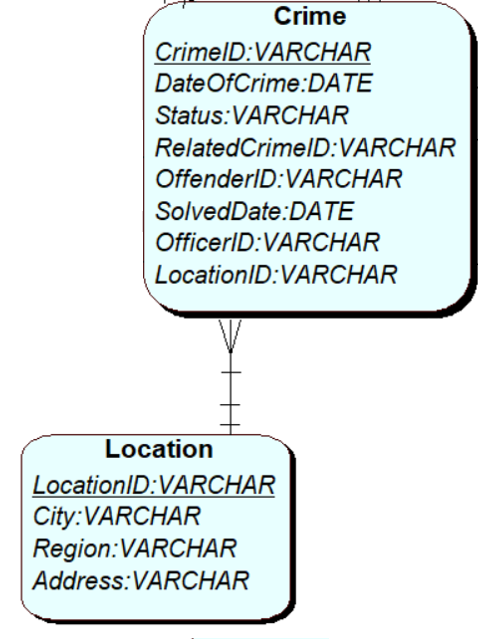
* Many to one relation of Crime to PoliceOfficer entity where foreign key lies at Crime entity.



Crime (CrimeID, DateOfCrime, Status, RelatedCrimeID, OffenderID, SolvedDate, *OfficerID*, LocationID)

PoliceOfficer (OfficerID, Name, ContactNo, InspectorID)

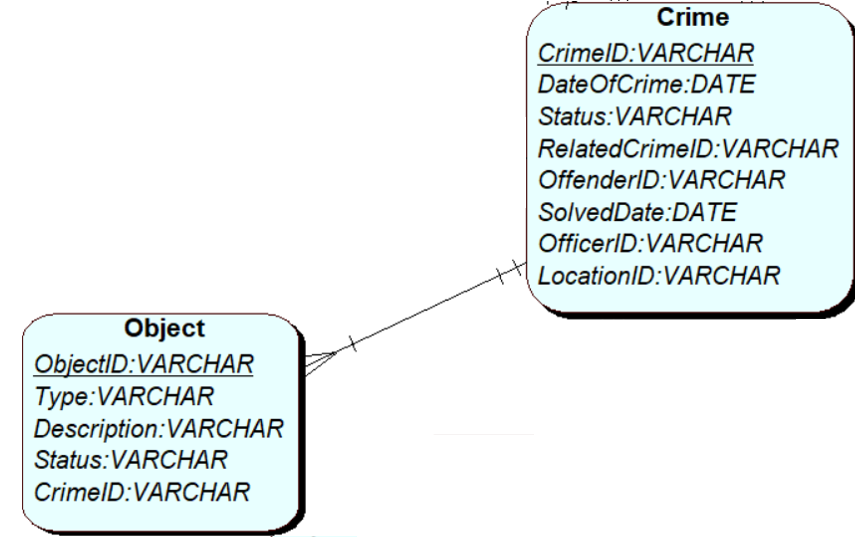
* Many to one relation of Crime to Location entity where foreign key lies at Crime entity.



Crime (CrimeID, DateOfCrime, Status, RelatedCrimeID, OffenderID, SolvedDate,OfficerID, *LocationID*)

LocationID (LocationID, City, Region, Address)

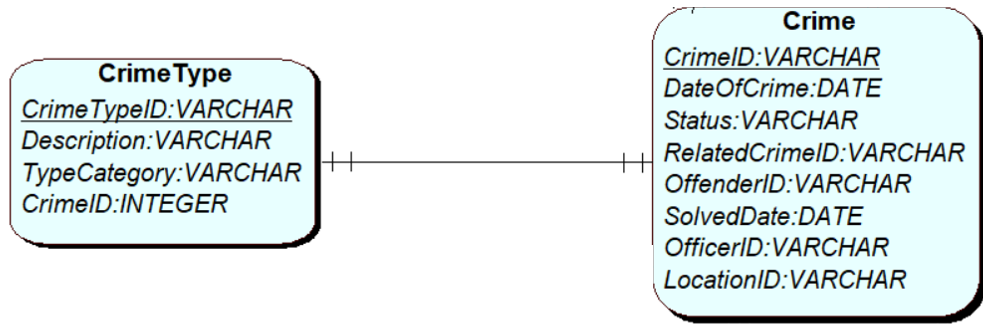
* One to many relation of crime with object entity where foreign key lies at Object entity.



Crime (CrimeID, DateOfCrime, Status, RelatedCrimeID, OffenderID, SolvedDate,OfficerID, LocationID)

# Object (ObjectID, Type, Description, Status, *CrimeID)*

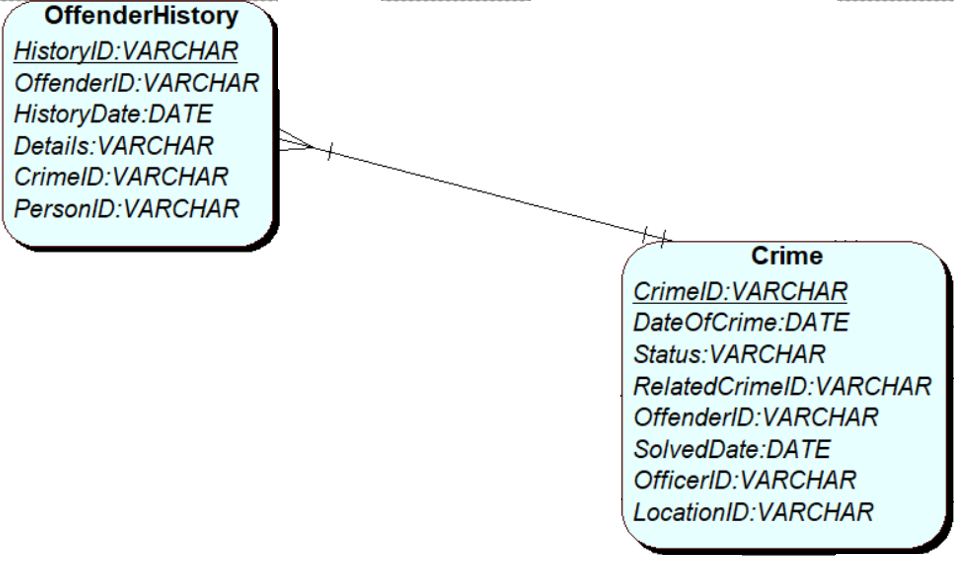
* One to one relation of Crime with CrimeType entity where foreign key can be placed at any entity



Crime (CrimeID, DateOfCrime, Status, RelatedCrimeID, OffenderID, SolvedDate,OfficerID, LocationID)

CrimeType (CrimeTypeID, Description, TypeCategory, *CrimeID*)

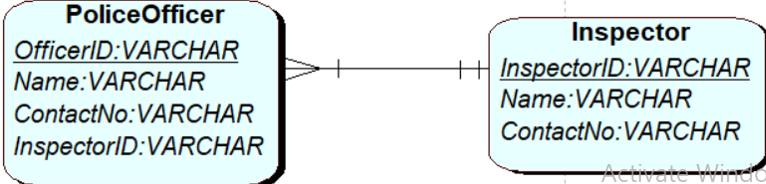
* One to many relation of Crime with OffenderHistory entity where foreign key at many end or at OffenderHistory.



Crime (CrimeID, DateOfCrime, Status, RelatedCrimeID, OffenderID, SolvedDate,OfficerID, LocationID)

OffenderHistory (HistoryID, OffenderID, HistoryDate, Details, *CrimeID,* PersonID)

* One to many relation of Inspector with PoliceOfficer entity so, foreign key at PoliceOfficer entity



PoliceOfficer (OfficerID, Name, ContactNo, *InspectorID*)

Inspector (InspectorID, Name, ContactNo)

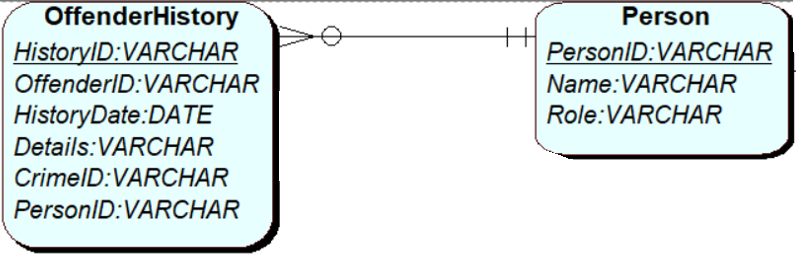
* One to many relation of Person with Person\_Event entity which is done to resolve many to many between person and event entity



Person (PersonID, Name, Role)

Person\_Event (EventID, PersonID)

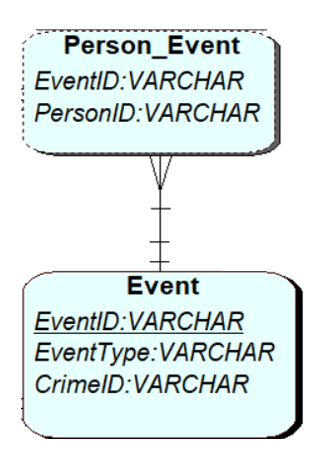
* One to many relation with optionality at many end of Person with OffenderHistory entity where foreign key at OffenderHistory.



OffenderHistory (HistoryID, OffenderID, HistoryDate, Details, CrimeID*, PersonID*)

Person (PersonID, Name, Role)

* One to many relation of Event with Person\_Event entity done in order to resolve many to many relation of person and event.



Person\_Event (EventID, PersonID)

Event (EventID, EventType, CrimeID)

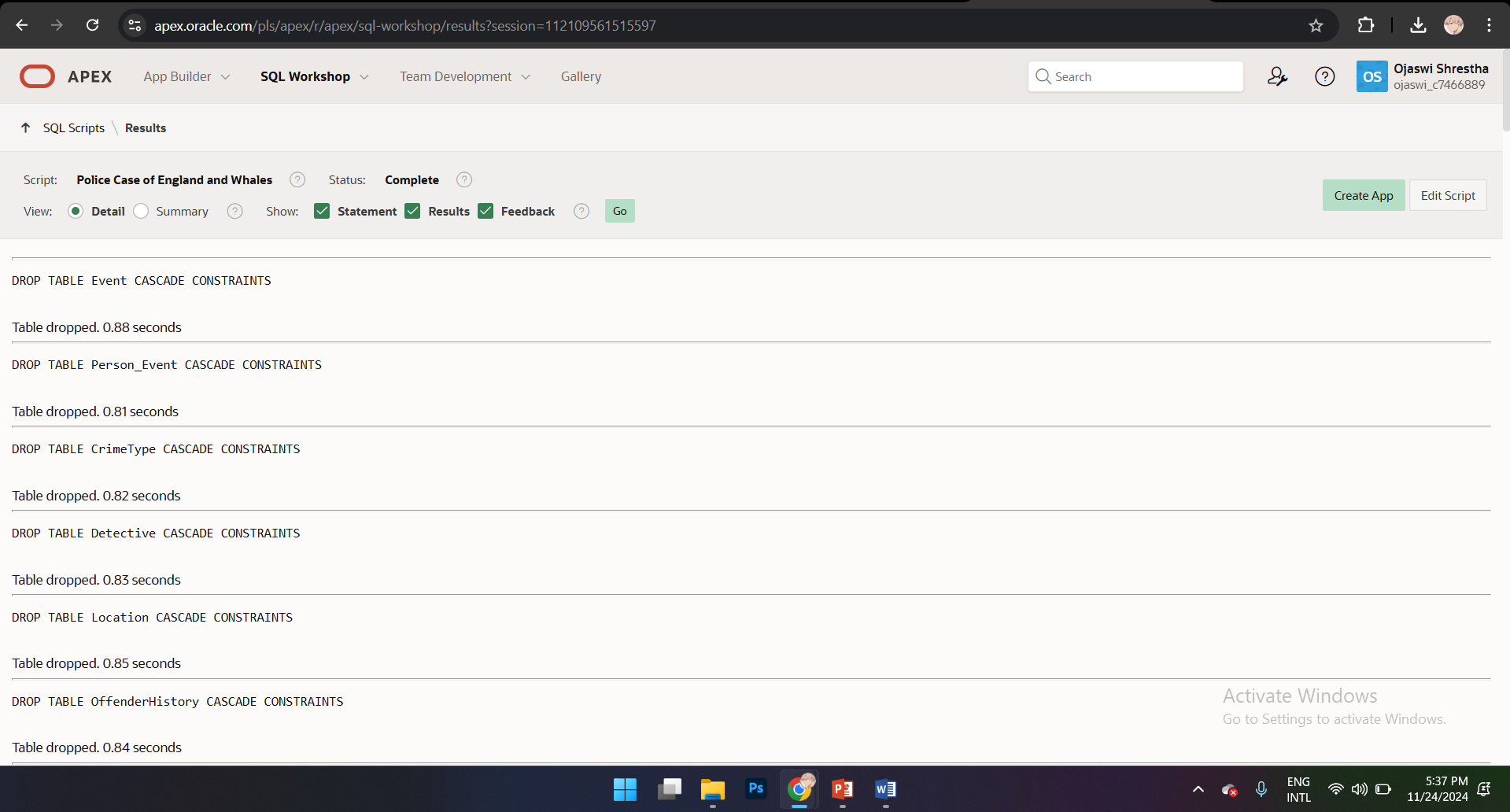
**Task 3: Physical Design**

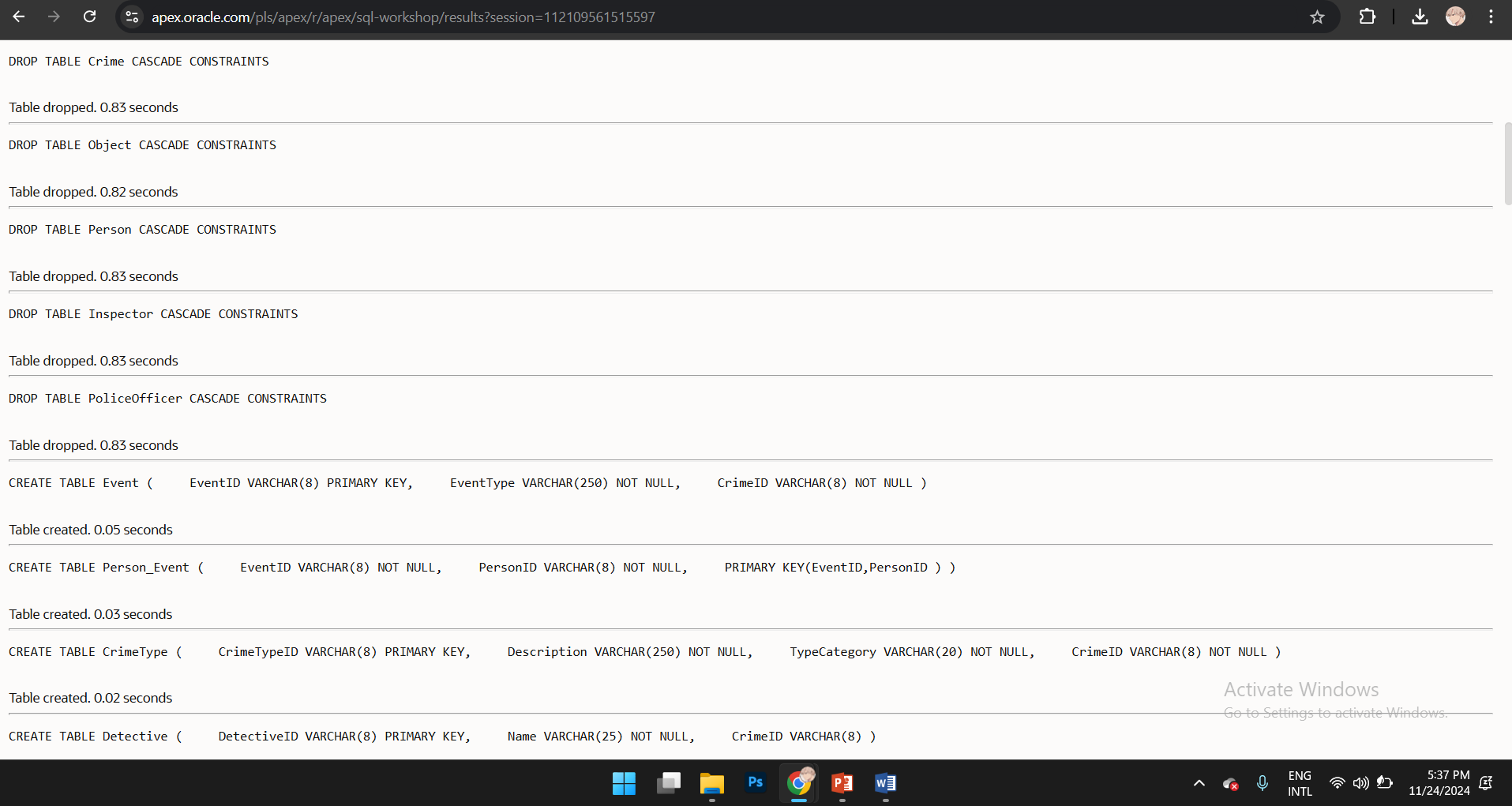
Physical design refers to the optimization of the table while implementing and ensuring the data integrity and avoiding data redundancy (IBM, n.d.).

* **Datatype and the sizes** 
  + - “VARCHAR” length should be different such as ids can be smaller than that of name and descriptions.
* **Constraints** 
  + - Adding “not null” is must in the fields such as in the attributes of name and descriptions
    - To have uniqueness while searching and sorting primary key to be enforced in the attributes such as in “CrimeID”, “HistoryID” etc.
* Partitioning
  + - For larger data set, partition can be done to improvise the query performance.

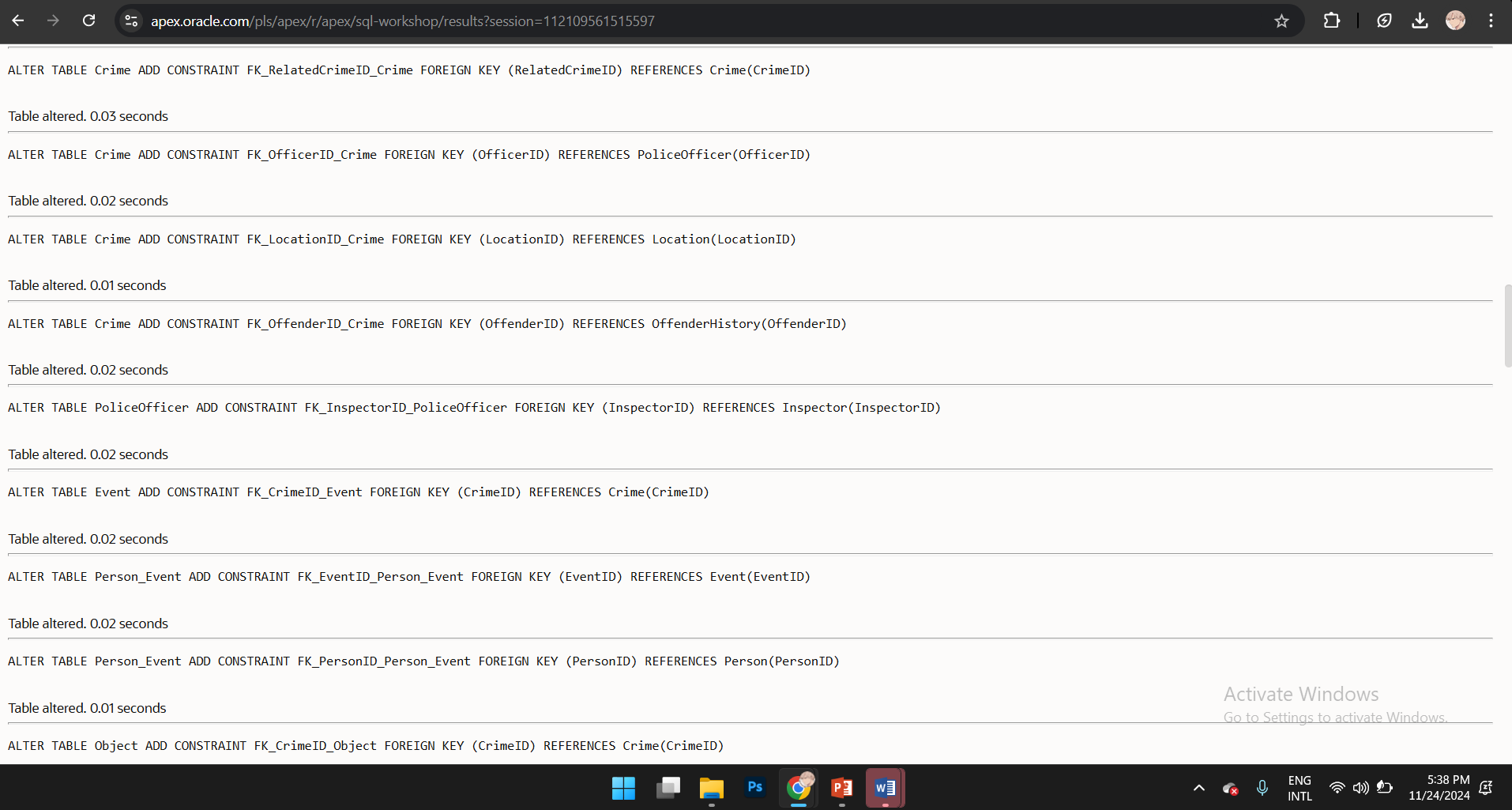
**Task 4: Tables implementation**

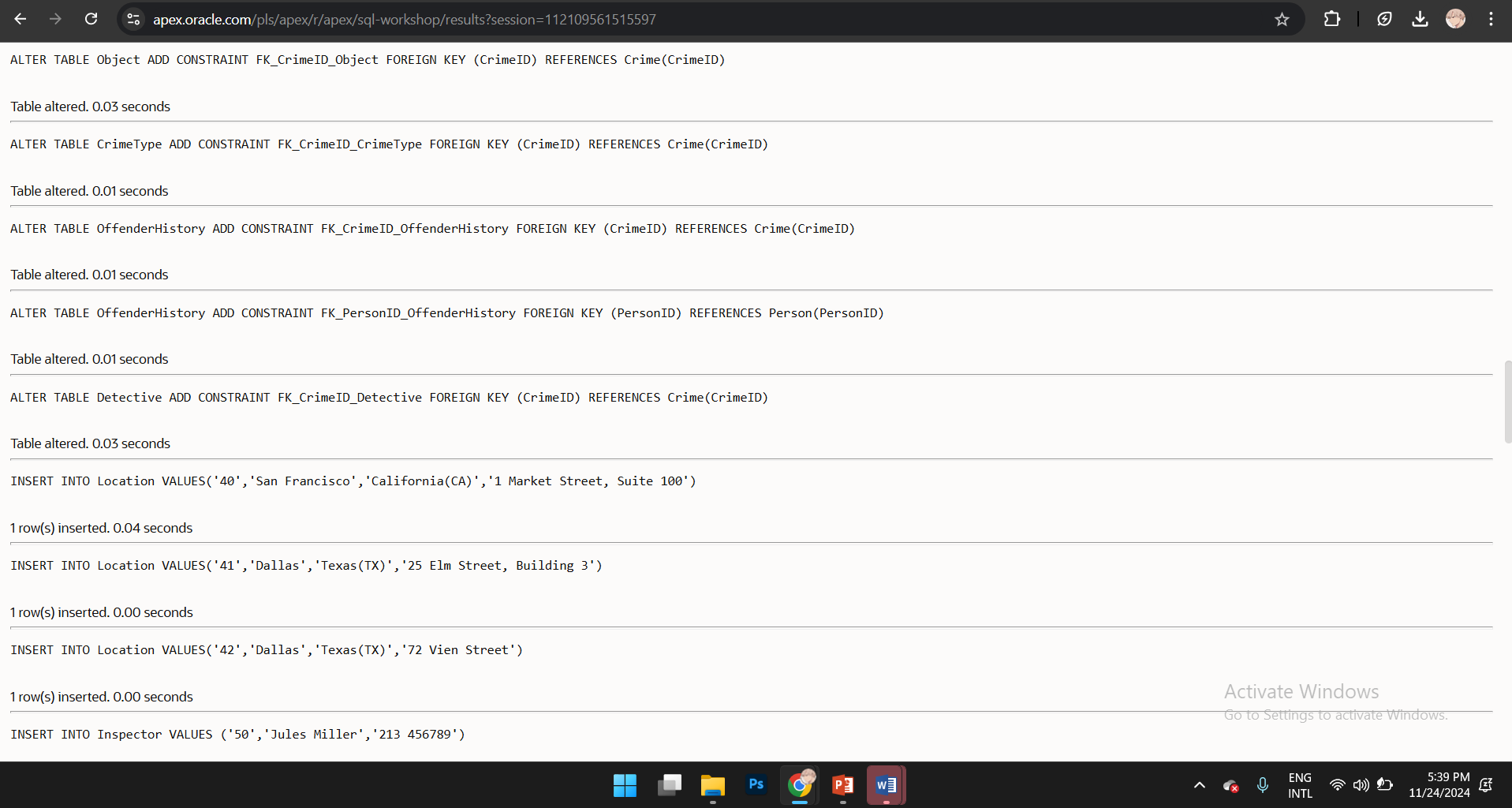
Detailed view:



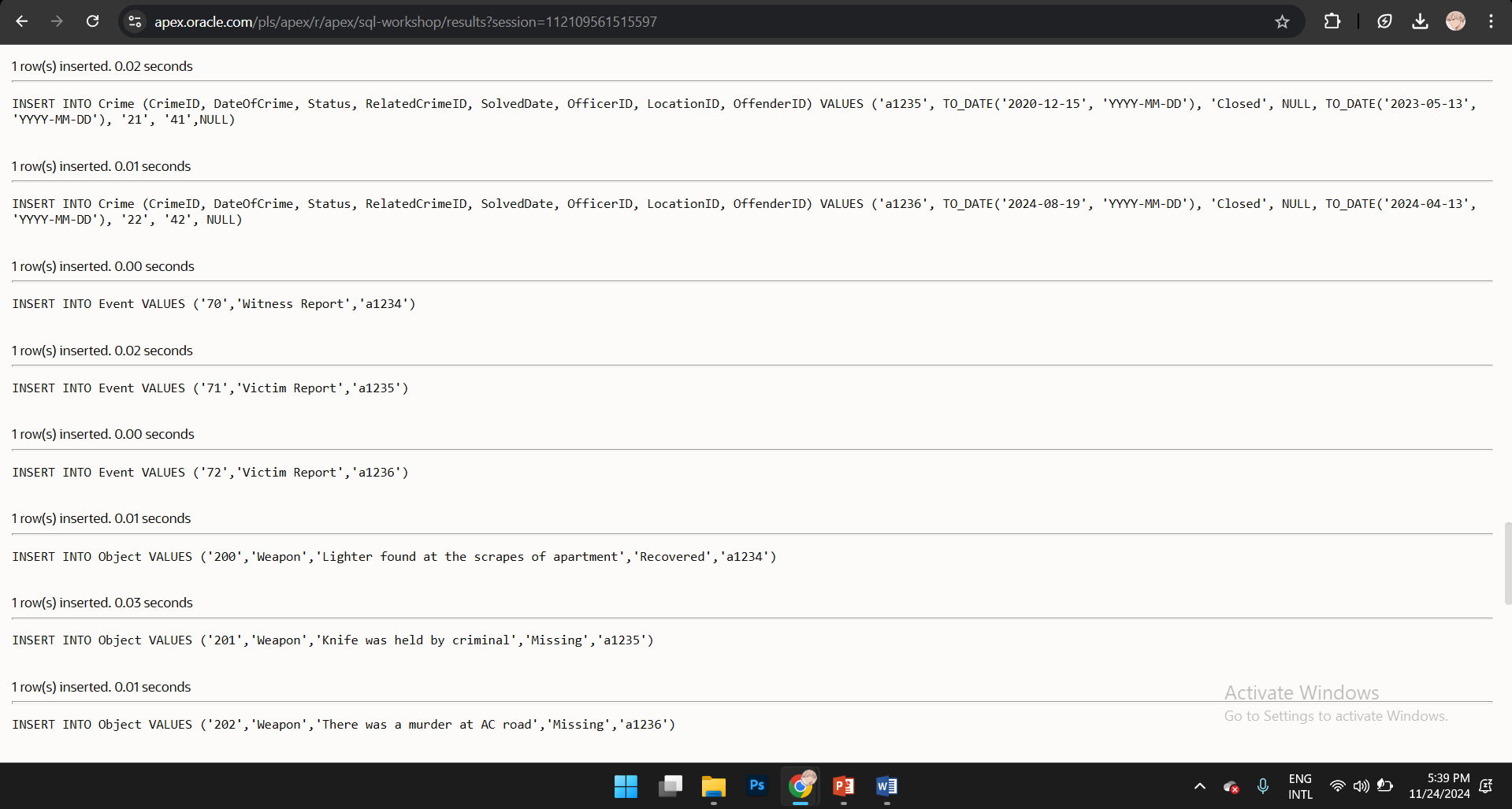


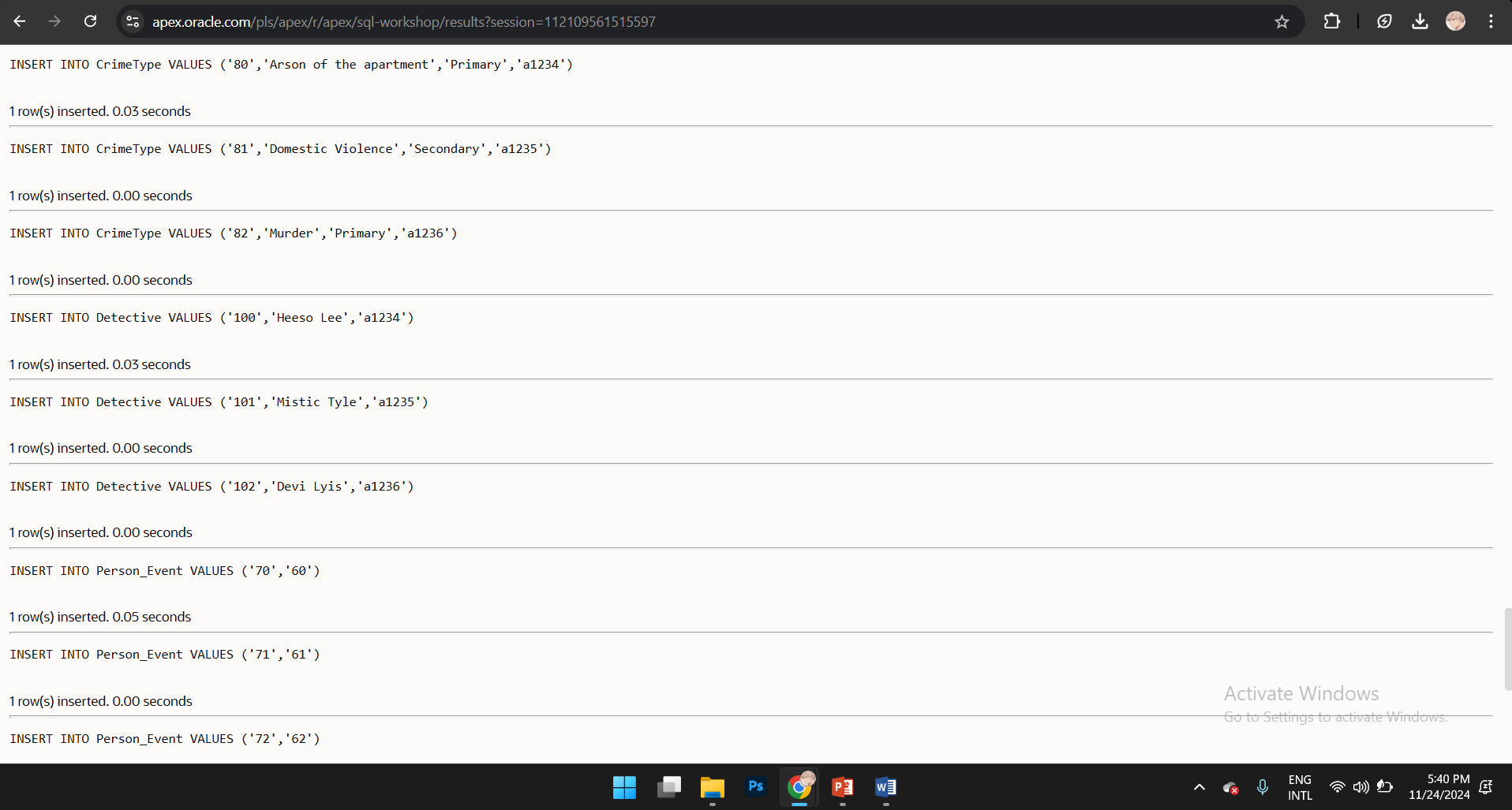


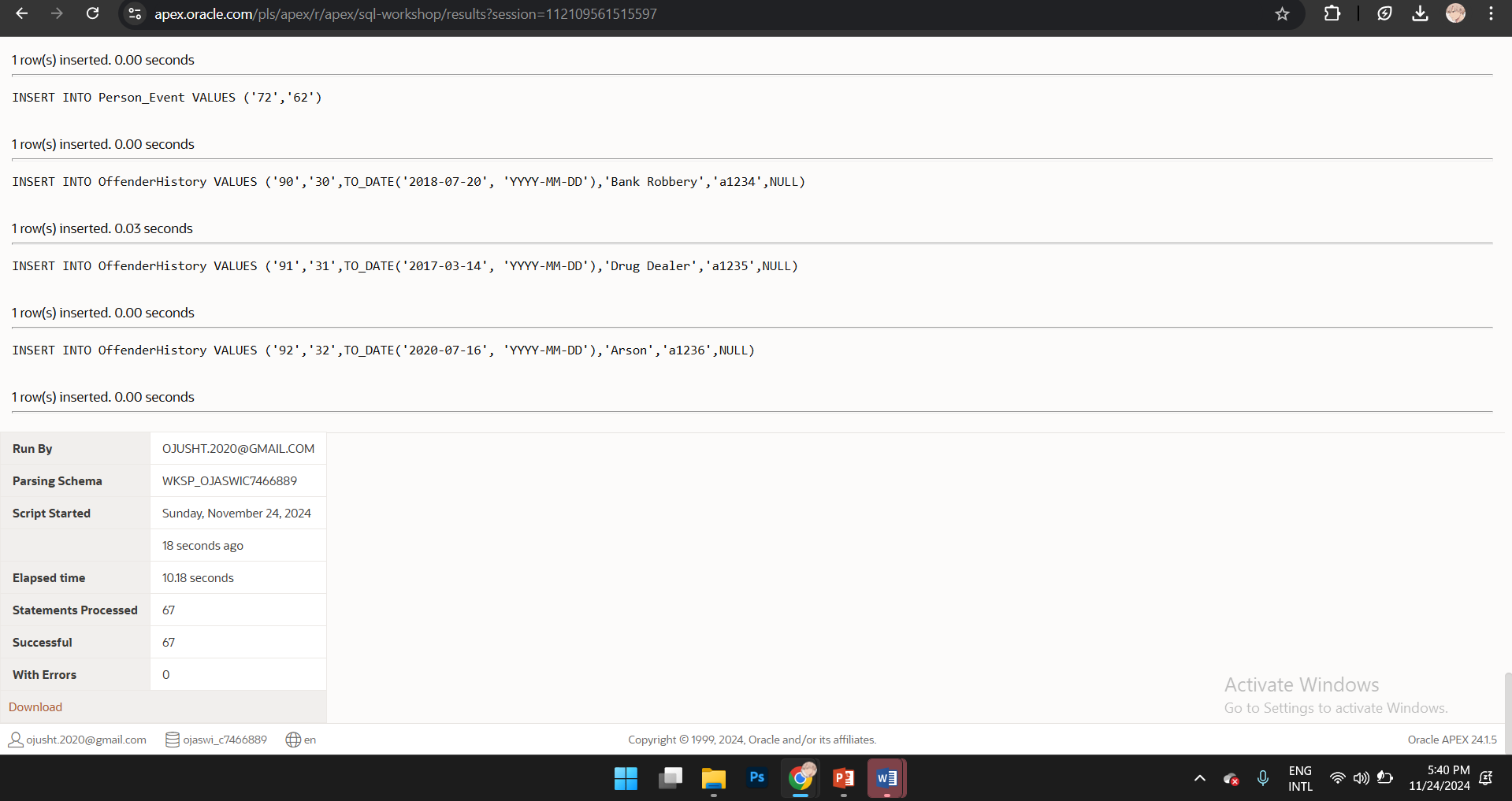




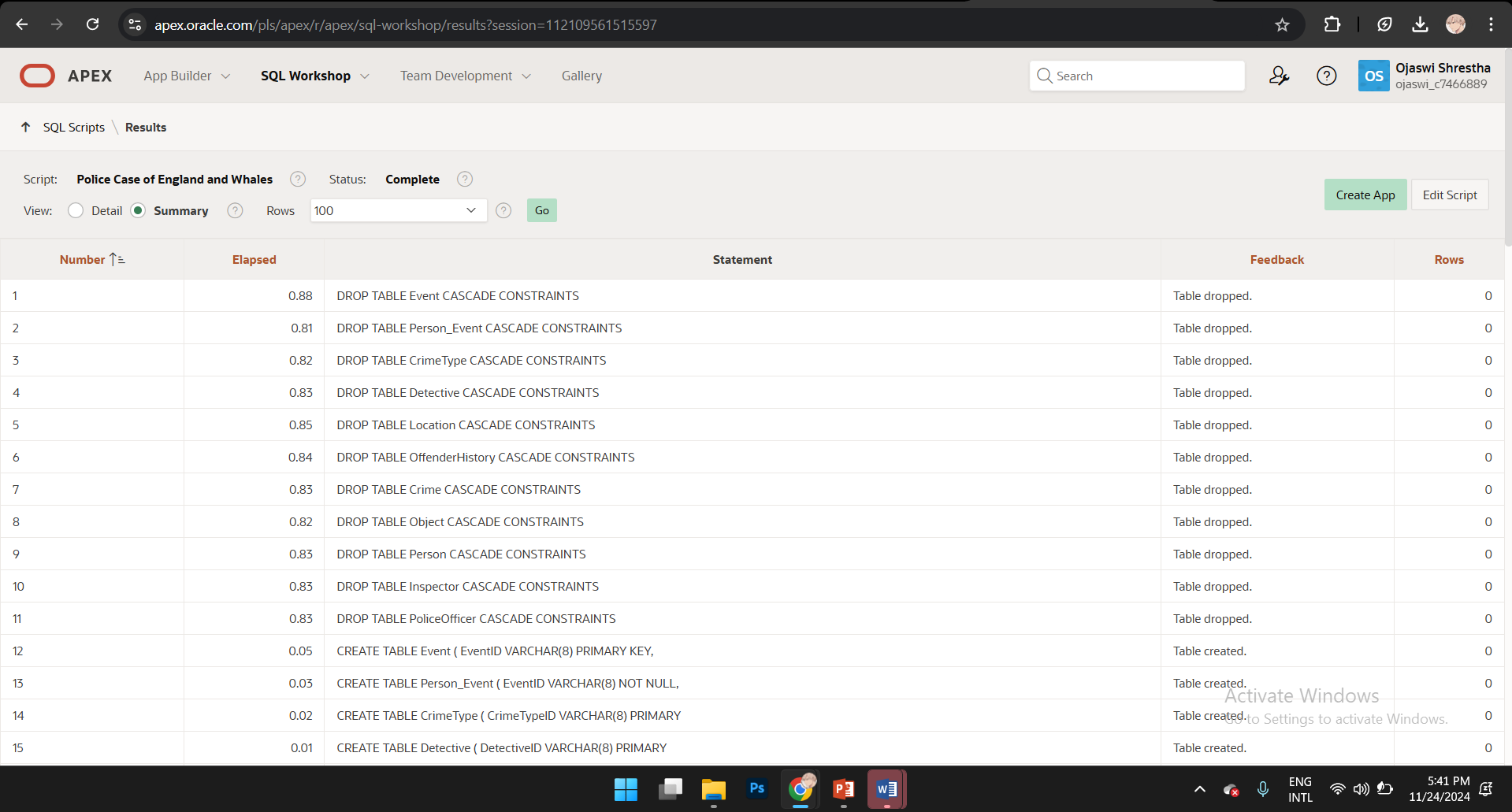


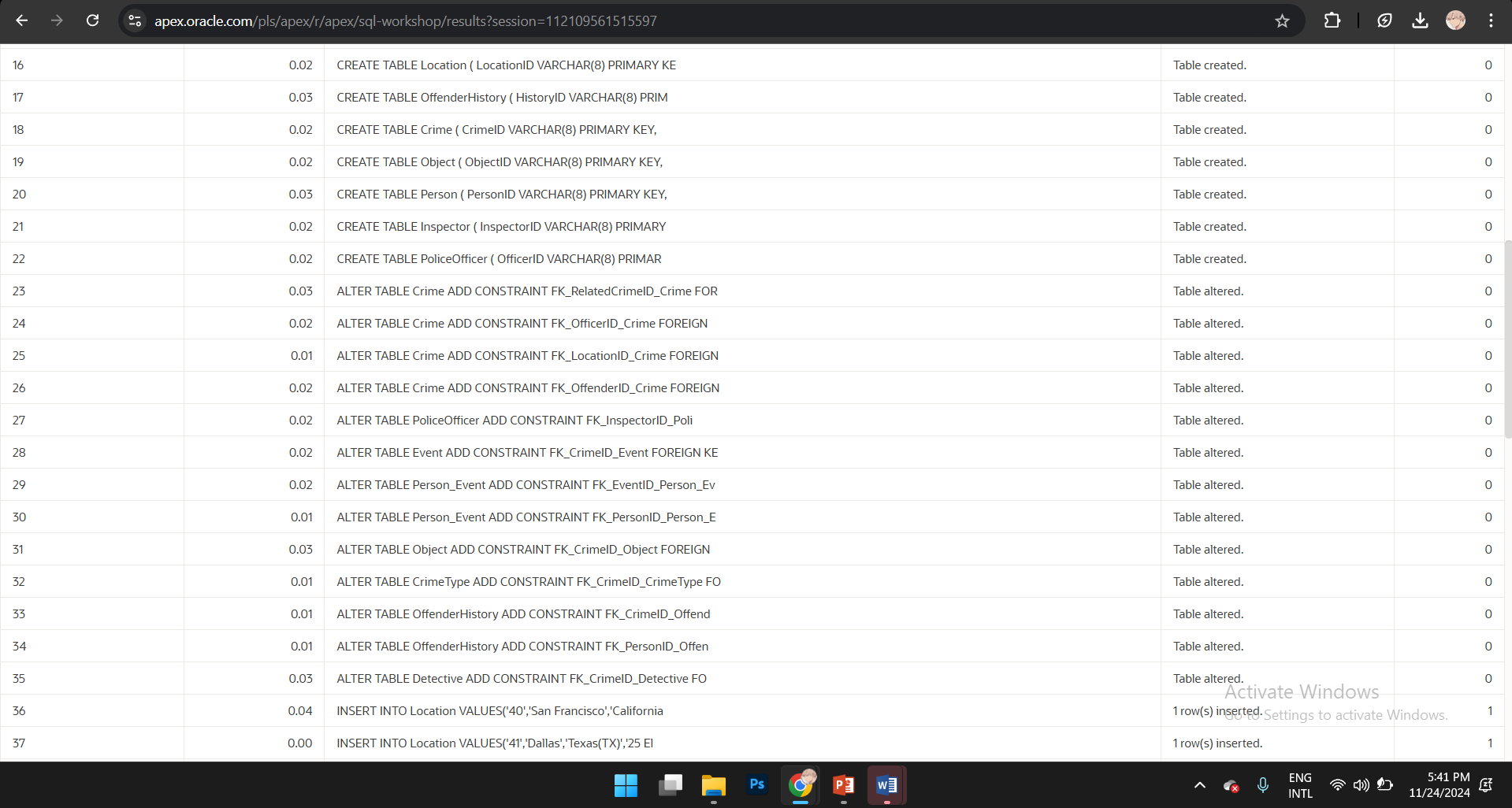


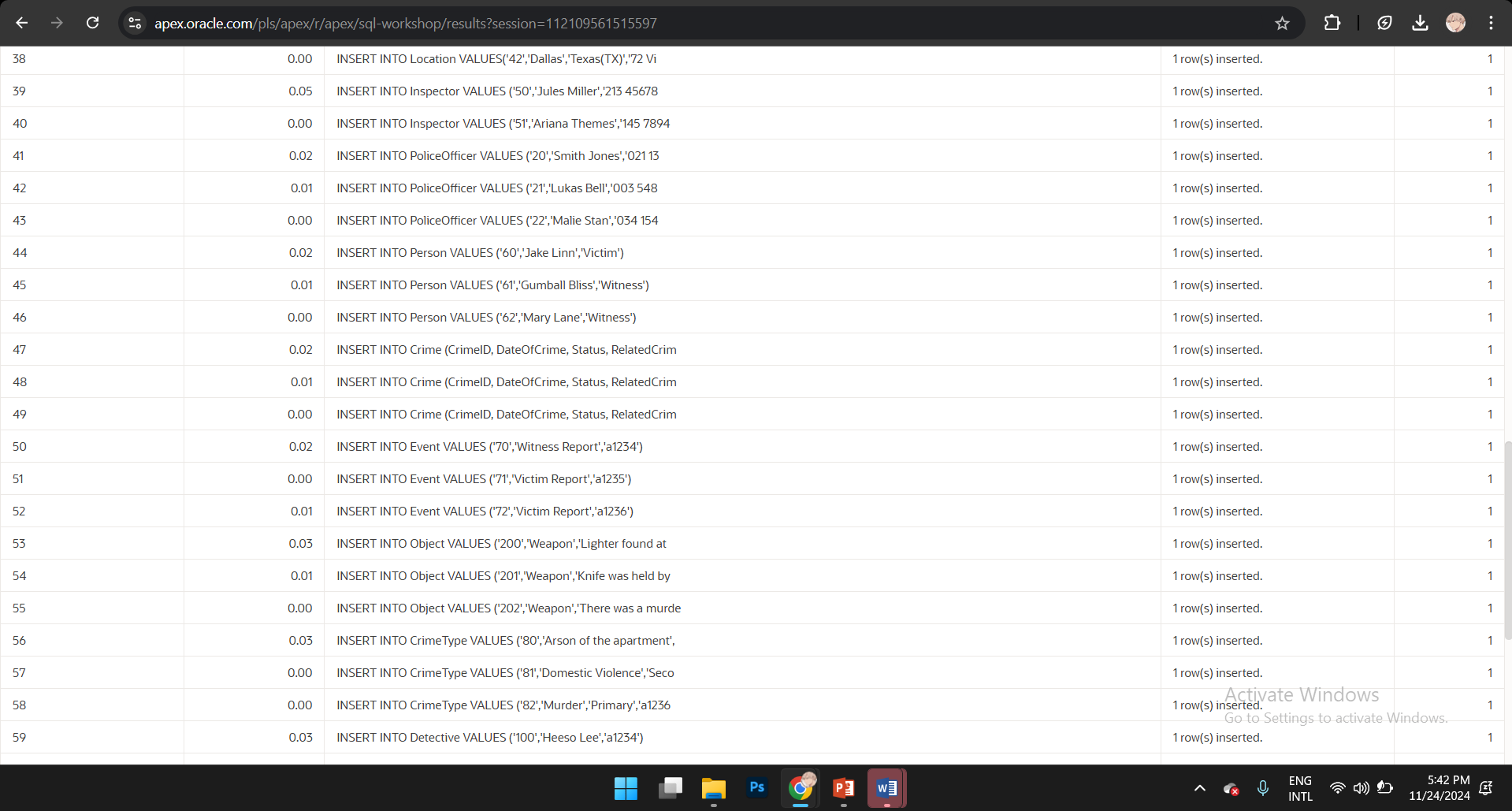


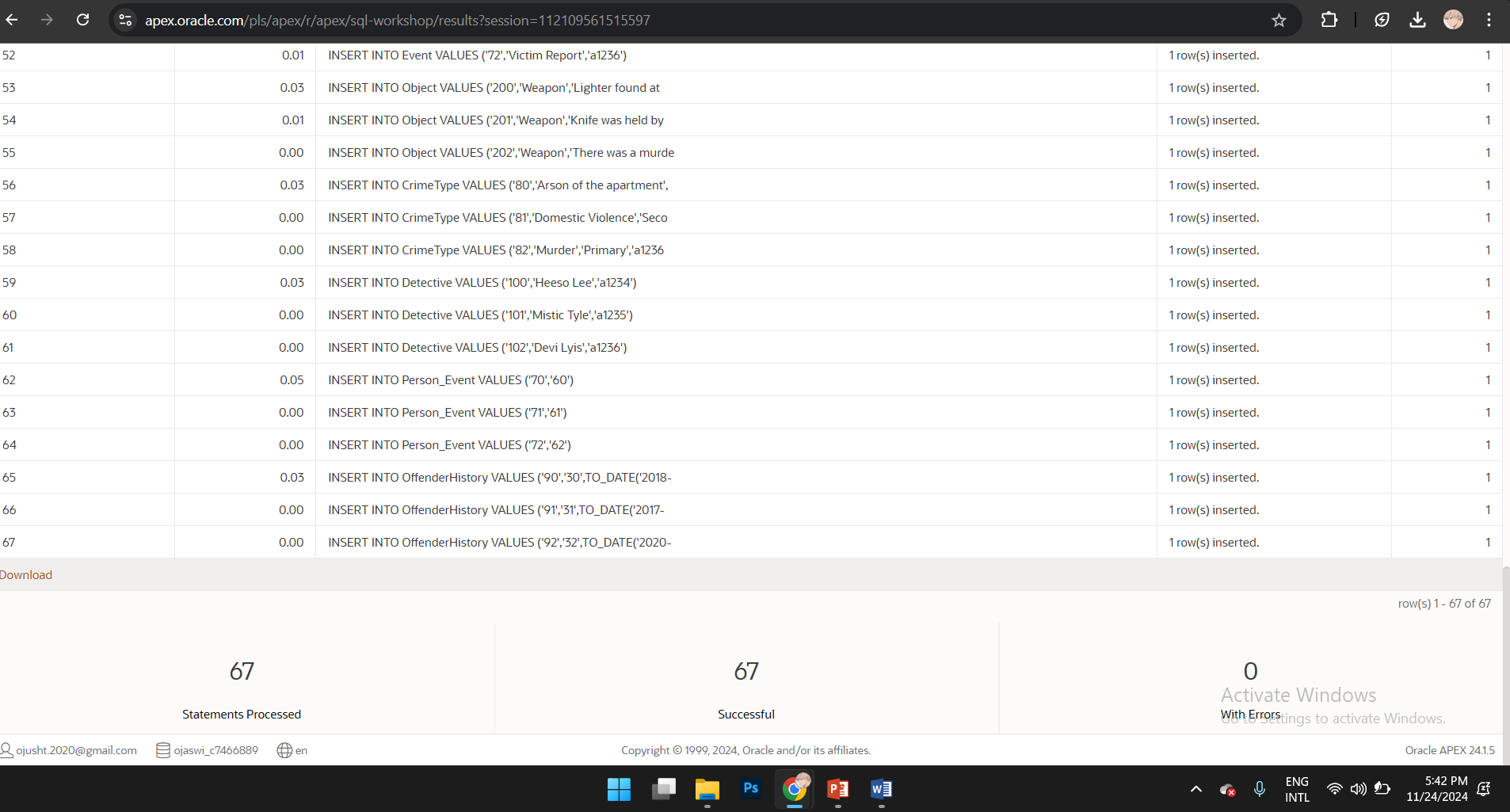


Summary screenshot:









SQL scripts:

-- Drop tables if they exist

DROP TABLE Event CASCADE CONSTRAINTS;

DROP TABLE Person\_Event CASCADE CONSTRAINTS;

DROP TABLE CrimeType CASCADE CONSTRAINTS;

DROP TABLE Detective CASCADE CONSTRAINTS;

DROP TABLE Location CASCADE CONSTRAINTS;

DROP TABLE OffenderHistory CASCADE CONSTRAINTS;

DROP TABLE Crime CASCADE CONSTRAINTS;

DROP TABLE Object CASCADE CONSTRAINTS;

DROP TABLE Person CASCADE CONSTRAINTS;

DROP TABLE Inspector CASCADE CONSTRAINTS;

DROP TABLE PoliceOfficer CASCADE CONSTRAINTS;

-- Create the "Event" table

CREATE TABLE Event (

EventID VARCHAR(8) PRIMARY KEY,

EventType VARCHAR(250) NOT NULL,

CrimeID VARCHAR(8) NOT NULL

);

-- Create the "Person\_Event" table TO RESOLVE MANY TO MANY RELATIONSHIP

CREATE TABLE Person\_Event (

EventID VARCHAR(8) NOT NULL,

PersonID VARCHAR(8) NOT NULL,

PRIMARY KEY(EventID,PersonID )

);

-- Create the "CrimeType" table

CREATE TABLE CrimeType (

CrimeTypeID VARCHAR(8) PRIMARY KEY,

Description VARCHAR(250) NOT NULL,

TypeCategory VARCHAR(20) NOT NULL,

CrimeID VARCHAR(8) NOT NULL

);

-- Create the "Detective" table

CREATE TABLE Detective (

DetectiveID VARCHAR(8) PRIMARY KEY,

Name VARCHAR(25) NOT NULL,

CrimeID VARCHAR(8)

);

-- Create the "Location" table

CREATE TABLE Location (

LocationID VARCHAR(8) PRIMARY KEY,

City VARCHAR(20) NOT NULL,

Region VARCHAR(20) NOT NULL,

Address VARCHAR(50) NOT NULL

);

-- Create the "OffenderHistory" table

CREATE TABLE OffenderHistory (

HistoryID VARCHAR(8) PRIMARY KEY,

OffenderID VARCHAR(8) NOT NULL UNIQUE,

HistoryDate DATE NOT NULL,

Details VARCHAR(250) NOT NULL,

CrimeID VARCHAR(8) NOT NULL,

PersonID VARCHAR(8)

);

-- Create the "Crime" table

CREATE TABLE Crime (

CrimeID VARCHAR(8) PRIMARY KEY,

DateOfCrime DATE NOT NULL,

Status VARCHAR(10) NOT NULL,

RelatedCrimeID VARCHAR(8),

SolvedDate DATE,

OfficerID VARCHAR(8),

LocationID VARCHAR(8),

OffenderID VARCHAR(8)

);

-- Create the "Object" table

CREATE TABLE Object (

ObjectID VARCHAR(8) PRIMARY KEY,

Type VARCHAR(15) NOT NULL,

Description VARCHAR(250) NOT NULL,

Status VARCHAR(10) NOT NULL,

CrimeID VARCHAR(8) NOT NULL

);

-- Create the "Person" table

CREATE TABLE Person (

PersonID VARCHAR(8) PRIMARY KEY,

Name VARCHAR(25) NOT NULL,

Role VARCHAR(10)

);

-- Create the "Inspector" table

CREATE TABLE Inspector (

InspectorID VARCHAR(8) PRIMARY KEY,

Name VARCHAR(25) NOT NULL,

ContactNo VARCHAR(15)

);

-- Create the "PoliceOfficer" table

CREATE TABLE PoliceOfficer (

OfficerID VARCHAR(8) PRIMARY KEY,

Name VARCHAR(20) NOT NULL,

ContactNo VARCHAR(15),

InspectorID VARCHAR(8) NOT NULL

);

-- adding foreign keys on the crime table

ALTER TABLE Crime

ADD CONSTRAINT FK\_RelatedCrimeID\_Crime FOREIGN KEY (RelatedCrimeID) REFERENCES Crime(CrimeID);

ALTER TABLE Crime

ADD CONSTRAINT FK\_OfficerID\_Crime FOREIGN KEY (OfficerID) REFERENCES PoliceOfficer(OfficerID);

ALTER TABLE Crime

ADD CONSTRAINT FK\_LocationID\_Crime FOREIGN KEY (LocationID) REFERENCES Location(LocationID);

ALTER TABLE Crime

ADD CONSTRAINT FK\_OffenderID\_Crime FOREIGN KEY (OffenderID) REFERENCES OffenderHistory(OffenderID);

-- ADDINF FOREIGN KEY IN PoliceOfficer

ALTER TABLE PoliceOfficer

ADD CONSTRAINT FK\_InspectorID\_PoliceOfficer FOREIGN KEY (InspectorID) REFERENCES Inspector(InspectorID);

-- ADDING FOREIGN KEY IN Event

ALTER TABLE Event

ADD CONSTRAINT FK\_CrimeID\_Event FOREIGN KEY (CrimeID) REFERENCES Crime(CrimeID);

-- adding foreign key to Person\_Event

ALTER TABLE Person\_Event

ADD CONSTRAINT FK\_EventID\_Person\_Event FOREIGN KEY (EventID) REFERENCES Event(EventID);

ALTER TABLE Person\_Event

ADD CONSTRAINT FK\_PersonID\_Person\_Event FOREIGN KEY (PersonID) REFERENCES Person(PersonID);

-- adding foreign keys on the Object table

ALTER TABLE Object

ADD CONSTRAINT FK\_CrimeID\_Object FOREIGN KEY (CrimeID) REFERENCES Crime(CrimeID);

-- adding foreign key on CrimeType table

ALTER TABLE CrimeType

ADD CONSTRAINT FK\_CrimeID\_CrimeType FOREIGN KEY (CrimeID) REFERENCES Crime(CrimeID);

-- adding foreign keys on the OffenderHistory table

ALTER TABLE OffenderHistory

ADD CONSTRAINT FK\_CrimeID\_OffenderHistory FOREIGN KEY (CrimeID) REFERENCES Crime(CrimeID);

ALTER TABLE OffenderHistory

ADD CONSTRAINT FK\_PersonID\_OffenderHistory FOREIGN KEY (PersonID) REFERENCES Person(PersonID);

-- adding foreign key in Detective table

ALTER TABLE Detective

ADD CONSTRAINT FK\_CrimeID\_Detective FOREIGN KEY (CrimeID) REFERENCES Crime(CrimeID);

--inserting data into location entity

INSERT INTO Location

VALUES('40','San Francisco','California(CA)','1 Market Street, Suite 100');

INSERT INTO Location

VALUES('41','Dallas','Texas(TX)','25 Elm Street, Building 3');

INSERT INTO Location

VALUES('42','Dallas','Texas(TX)','72 Vien Street');

--inserting data into Inspector entity

INSERT INTO Inspector

VALUES ('50','Jules Miller','213 456789');

INSERT INTO Inspector

VALUES ('51','Ariana Themes','145 789456');

--inserting data into PoliceOfficer entity

INSERT INTO PoliceOfficer

VALUES ('20','Smith Jones','021 136587','50');

INSERT INTO PoliceOfficer

VALUES ('21','Lukas Bell','003 548632','51');

INSERT INTO PoliceOfficer

VALUES ('22','Malie Stan','034 154687','51');

--inserting data into Person entity

INSERT INTO Person

VALUES ('60','Jake Linn','Victim');

INSERT INTO Person

VALUES ('61','Gumball Bliss','Witness');

INSERT INTO Person

VALUES ('62','Mary Lane','Witness');

--inserting data into crime entity

INSERT INTO Crime (CrimeID, DateOfCrime, Status, RelatedCrimeID, SolvedDate, OfficerID, LocationID, OffenderID)

VALUES ('a1234', TO\_DATE('2020-11-26', 'YYYY-MM-DD'), 'Open', NULL, NULL,'20', '40', NULL);

INSERT INTO Crime (CrimeID, DateOfCrime, Status, RelatedCrimeID, SolvedDate, OfficerID, LocationID, OffenderID)

VALUES ('a1235', TO\_DATE('2020-12-15', 'YYYY-MM-DD'), 'Closed', NULL, TO\_DATE('2023-05-13', 'YYYY-MM-DD'), '21', '41',NULL);

INSERT INTO Crime (CrimeID, DateOfCrime, Status, RelatedCrimeID, SolvedDate, OfficerID, LocationID, OffenderID)

VALUES ('a1236', TO\_DATE('2024-08-19', 'YYYY-MM-DD'), 'Closed', NULL, TO\_DATE('2024-04-13', 'YYYY-MM-DD'), '22', '42', NULL);

--inserting data into Event entity

INSERT INTO Event

VALUES ('70','Witness Report','a1234');

INSERT INTO Event

VALUES ('71','Victim Report','a1235');

INSERT INTO Event

VALUES ('72','Victim Report','a1236');

--inserting data into Object entity

INSERT INTO Object

VALUES ('200','Weapon','Lighter found at the scrapes of apartment','Recovered','a1234');

INSERT INTO Object

VALUES ('201','Weapon','Knife was held by criminal','Missing','a1235');

INSERT INTO Object

VALUES ('202','Weapon','There was a murder at AC road','Missing','a1236');

--inserting data into CrimeType entity

INSERT INTO CrimeType

VALUES ('80','Arson of the apartment','Primary','a1234');

INSERT INTO CrimeType

VALUES ('81','Domestic Violence','Secondary','a1235');

INSERT INTO CrimeType

VALUES ('82','Murder','Primary','a1236');

--inserting data into Detective entity

INSERT INTO Detective

VALUES ('100','Heeso Lee','a1234');

INSERT INTO Detective

VALUES ('101','Mistic Tyle','a1235');

INSERT INTO Detective

VALUES ('102','Devi Lyis','a1236');

--inserting data into Person\_Event entity

INSERT INTO Person\_Event

VALUES ('70','60');

INSERT INTO Person\_Event

VALUES ('71','61');

INSERT INTO Person\_Event

VALUES ('72','62');

--inserting data into OffenderHistory entity

INSERT INTO OffenderHistory

VALUES ('90','30',TO\_DATE('2018-07-20', 'YYYY-MM-DD'),'Bank Robbery','a1234',NULL);

INSERT INTO OffenderHistory

VALUES ('91','31',TO\_DATE('2017-03-14', 'YYYY-MM-DD'),'Drug Dealer','a1235',NULL);

INSERT INTO OffenderHistory

VALUES ('92','32',TO\_DATE('2020-07-16', 'YYYY-MM-DD'),'Arson','a1236',NULL);

# 5. Bibliography

* IBM, n.d., Physical database dasign, IBM Documentation. Available at

<https://www.ibm.com/docs/en/db2-for-zos/12?topic=relationships-physical-database-design> [Accessed 24 November 2024].

**GENERATIVE AI USE**

|  |  |  |  |
| --- | --- | --- | --- |
| **Numbering** | **Generative AI Tool (e.g. ChatGPT)** | **How generative AI Tool was used** | **Reference** |
| 1 | ChatGPT 4 | Asked for the introduction to report | OpenAI. (2024). ChatGPT (Feb 22 version) [Large language model]. <https://chat.openai.com/chat>  Introduction to the Report  In today’s rapidly evolving digital age, businesses and organizations heavily rely on robust data storage and management systems to perform their operations efficiently and effectively. |