```
-- Chapter 4

-- Dropped tables

DROP TABLE Appeals;

DROP TABLE Crime_officers;

DROP TABLE Crime_charges;



-- Same tables as chapter 3 to be altered to include constraints
-- specified in chap 3 and extra constraints (and default values)
-- that I feel are needed.

-- *NOTE*
-- NOT NULL(s) were placed with all check constraints as the check constraint
-- allows the value NULL to be stored(assuming the condition doesn't invlove
-- null). Same with foreign keys.

-- DEFAULT alters gave me syntax errors when I attempted to put a
-- CONSTRAINT name on it. For this reason I removed the constraint name.

------------------------------------------------------------

-- Criminals table alterations

-- Criminal_ID primary key
ALTER TABLE Criminals
```

```
  ADD CONSTRAINT criminals_criminal_id_pk PRIMARY KEY(Criminal_ID);

-- Last not null -> Important to ensure there is a name associated with an ID
ALTER TABLE Criminals
  MODIFY ("Last" CONSTRAINT criminals_last_nn NOT NULL);

-- First not null -> Important to ensure there is a name associated with an ID
ALTER TABLE Criminals
  MODIFY ("First" CONSTRAINT criminals_first_nn NOT NULL);

-- V_status check
ALTER TABLE Criminals
  ADD CONSTRAINT criminals_v_status_ck CHECK(V_status = 'N'
    AND V_status = 'Y');

-- V_status not null
ALTER TABLE Criminals
  MODIFY (V_status CONSTRAINT criminals_v_status_nn NOT NULL);

-- P_status check
ALTER TABLE Criminals
  ADD CONSTRAINT criminals_p_status_ck CHECK(P_status = 'N'
    AND P_status = 'Y');

-- P_status not null
ALTER TABLE Criminals
  MODIFY (P_status CONSTRAINT criminals_p_status_nn NOT NULL);

-------------------------------------------------------------
```

```
-- Aliases table alterations

-- Alias_ID primary key
ALTER TABLE Aliases
  ADD CONSTRAINT aliases_alias_id_pk PRIMARY KEY(Alias_ID);

-- Criminal_ID foreign key -> associates an alias with a criminal
ALTER TABLE Aliases
  ADD CONSTRAINT aliases_criminal_id_fk FOREIGN KEY(Criminal_ID)
    REFERENCES Criminals (Criminal_ID);

-- Criminal_ID not null
ALTER TABLE Aliases
  MODIFY (Criminal_ID CONSTRAINT aliases_criminal_id_nn NOT NULL);

-- Alias not null -> must have an alias associated with alias id.
ALTER TABLE Aliases
  MODIFY ("Alias" CONSTRAINT aliases_alias_nn NOT NULL);

------------------------------------------------------------

-- Crimes table alterations

-- Crime_ID primary key
ALTER TABLE Crimes
  ADD CONSTRAINT crimes_crime_id_pk PRIMARY KEY(Crime_ID);

-- Criminal_ID foreign key -> associates an alias with a crime
```

```
ALTER TABLE Crimes
  ADD CONSTRAINT crimes_criminal_id_fk FOREIGN KEY(Criminal_ID)
    REFERENCES Criminals (Criminal_ID);

-- Criminal_ID not null
ALTER TABLE Crimes
  MODIFY (Criminal_ID CONSTRAINT crimes_criminal_id_nn NOT NULL);

-- Classification check
ALTER TABLE Crimes
  ADD CONSTRAINT crimes_classification_ck
    CHECK (Classification = 'F' OR Classification = 'M'
    OR Classification = 'O' OR Classification = 'U');

-- Classification NOT NULL
ALTER TABLE Crimes
  MODIFY (Classification CONSTRAINT crimes_classification_nn NOT NULL);


-- Date_charged not null
ALTER TABLE Crimes
  MODIFY (Date_charged CONSTRAINT crimes_date_charged_nn NOT NULL);

-- Status check
ALTER TABLE Crimes
  ADD CONSTRAINT crimes_status_ck
    CHECK (Status = 'CL' OR Status = 'CA' OR Status = 'LA');

-- Status not null
```

```
ALTER TABLE Crimes
  MODIFY (Status CONSTRAINT crimes_status_nn NOT NULL);

-- Hearing_date check -> ensures this is before or at the same time as the
-- date the criminal was charged
ALTER TABLE Crimes
  ADD CONSTRAINT crimes_hearing_date_ck
    CHECK (Hearing_date <= Date_charged);

-- Hearing_date not null -> all crimes must have a court case
ALTER TABLE Crimes
  MODIFY (Hearing_date CONSTRAINT crimes_hearing_date_nn NOT NULL);

-- Appeal_cut_date -> computed coloumn, appeal cut off date is always 60 days
-- after hearing date.
-- I added a Virtual coloumn to achieve this.
ALTER TABLE Crimes
  ADD (Appeal_cut_date AS (hearing_date + 60));

-- Date_recorded not null
ALTER TABLE Crimes
  MODIFY (Date_recorded CONSTRAINT crimes_date_recorded_nn NOT NULL);

-------------------------------------------------------------

-- Prob_officers alterations
-- Important to have all the information of an officer as they are working.
-- Thus all fields are NOT NULL.
```

```
-- Prob_ID primary key
ALTER TABLE Prob_officers
  ADD CONSTRAINT prob_officers_prob_id_pk PRIMARY KEY(Prob_ID);

-- Last not null
ALTER TABLE Prob_officers
  MODIFY ("Last" CONSTRAINT prob_officers_last_nn NOT NULL);

-- First not null
ALTER TABLE Prob_officers
  MODIFY ("First" CONSTRAINT prob_officers_first_nn NOT NULL);

-- Street not null
ALTER TABLE Prob_officers
  MODIFY (Street CONSTRAINT prob_officers_street_nn NOT NULL);

-- City not null
ALTER TABLE Prob_officers
  MODIFY (City CONSTRAINT prob_officers_city_nn NOT NULL);

-- State not null
ALTER TABLE Prob_officers
  MODIFY ("State" CONSTRAINT prob_officers_state_nn NOT NULL);

-- Zip not null
ALTER TABLE Prob_officers
  MODIFY (Zip CONSTRAINT prob_officers_zip_nn NOT NULL);

-- Pager# not null
```

```
ALTER TABLE Prob_officers
  MODIFY (Pager# CONSTRAINT prob_officers_pager#_nn NOT NULL);


-- No police officer should have the same contact information as another
-- police officer.

-- Pager# unique
ALTER TABLE prob_officers
  ADD CONSTRAINT prob_officers_pager#_uk UNIQUE(Pager#);

-- Phone not null
ALTER TABLE Prob_officers
  MODIFY (Phone CONSTRAINT prob_officers_phone_nn NOT NULL);

-- Phone unique
ALTER TABLE prob_officers
  ADD CONSTRAINT prob_officers_phone_uk UNIQUE(Phone);

-- Email not null
ALTER TABLE Prob_officers
  MODIFY (Email CONSTRAINT prob_officers_email_n NOT NULL);

-- Email unique
ALTER TABLE Prob_officers
  ADD CONSTRAINT prob_officers_email_uk UNIQUE(Email);

-- Status check
ALTER TABLE Prob_officers
```

```
  ADD CONSTRAINT prob_officers_status_ck
    CHECK (Status = 'A' OR Status = 'I');

ALTER TABLE Prob_officers
  MODIFY (Status CONSTRAINT prob_officer_status_nn NOT NULL);

-------------------------------------------------------------

-- Sentences table alterations

-- Sentence_ID primary key
Alter TABLE Sentences
  ADD CONSTRAINT sentences_sentence_id_pk PRIMARY KEY(Sentence_ID);

-- Criminal_ID foreign key
ALTER TABLE Sentences
  ADD CONSTRAINT sentences_criminal_id_fk FOREIGN KEY(Criminal_ID)
    REFERENCES Criminals (Criminal_ID);

-- Criminal_ID not null
ALTER TABLE Sentences
  MODIFY (Criminal_ID CONSTRAINT sentences_criminal_id_nn NOT NULL);

-- Type_crime check
ALTER TABLE Sentences
  ADD CONSTRAINT sentences_type_ck
    CHECK ("Type" = 'J'
      OR "Type" = 'H' OR "Type" = 'P');
```

```
-- Type NOT NULL
ALTER TABLE Sentences
  MODIFY ("Type" CONSTRAINT sentences_type_nn NOT NULL);

-- Prob_ID foreign key
ALTER TABLE Sentences
  ADD CONSTRAINT sentences_prob_id_fk FOREIGN KEY(Prob_ID)
    REFERENCES Prob_officers(Prob_ID);

-- Prob_ID not null
ALTER TABLE Sentences
  MODIFY (Prob_ID CONSTRAINT sentences_prob_id_nn NOT NULL);

-- Start_date check -> ensures starting date of a sentence is always before
-- the end of a sentence.
ALTER TABLE Sentences
  ADD CONSTRAINT sentences_start_date_ck
    CHECK (Start_date < End_date);

-- Start_date not null
ALTER TABLE Sentences
  MODIFY (Start_date CONSTRAINT sentences_start_date_nn NOT NULL);

-- Start_date default
ALTER TABLE Sentences
  MODIFY (Start_date DEFAULT SYSDATE);

-- End_date check -> ensures ending date of a senetence is always after
-- the start of the sentence
```

```
ALTER TABLE Sentences
  ADD CONSTRAINT sentences_end_date_ck
    CHECK (End_date > Start_date);

-- End_date not null
ALTER TABLE Sentences
  MODIFY (End_date CONSTRAINT sentences_end_date_nn NOT NULL);

-- Violations check -> cannot have a negative amount of violations
ALTER TABLE Sentences
  ADD CONSTRAINT sentences_violations_ck
    CHECK (Violations >= 0);

-- Violations NOT NULL
ALTER TABLE Sentences
  MODIFY (Violations CONSTRAINT senetences_violations_nn NOT NULL);

------------------------------------------------------------

-- Officers alterations
-- Important to have all the information of an officer as they are working.
-- Thus all fields are NOT NULL.

-- Officer_ID primary key
ALTER TABLE Officers
  ADD CONSTRAINT officers_officer_id_pk PRIMARY KEY(Officer_ID);

-- Last not null
ALTER TABLE Officers
```

```
  MODIFY ("Last" CONSTRAINT officers_last_nn NOT NULL);

-- First not null
ALTER TABLE Officers
  MODIFY ("First" CONSTRAINT officers_first_nn NOT NULL);

-- Precinct not null
ALTER TABLE Officers
  MODIFY (Precinct CONSTRAINT officers_precinct_nn NOT NULL);

-- Badge not null
ALTER TABLE Officers
  MODIFY (Badge CONSTRAINT officers_badge_nn NOT NULL);

-- Badge unique
ALTER TABLE Officers
  ADD CONSTRAINT officers_badge_uk UNIQUE(Badge);


-- No police officer should have the same contact information as another
-- police officer.

-- Phone not null
ALTER TABLE Officers
  MODIFY (Phone CONSTRAINT officers_phone_nn NOT NULL);

-- Phone unique
ALTER TABLE Officers
  ADD CONSTRAINT officers_phone_uk UNIQUE(Phone);
```

```
-- Status check
ALTER TABLE Officers
  ADD CONSTRAINT officers_status_ck
    CHECK (Status = 'A' OR Status = 'I');

-- Status NOT NULL
ALTER TABLE Officers
  MODIFY (Status CONSTRAINT officers_status_nn NOT NULL);

------------------------------------------------------------

-- Crime_codes alterations

-- Crime_code primary key
ALTER TABLE Crime_codes
  ADD CONSTRAINT crime_codes_crime_code_pk PRIMARY KEY(Crime_code);

-- Code_description -> important to have a description for every crime code.
ALTER TABLE Crime_codes
  MODIFY (Code_description CONSTRAINT crime_codes_code_nn NOT NULL);

------------------------------------------------------------

-- Appeals recreation

CREATE TABLE Appeals
  (Appeal_ID NUMBER(5),
  Crime_ID NUMBER(9, 0) NOT NULL,
```

```
  Filing_date DATE DEFAULT SYSDATE NOT NULL,
  Hearing_date DATE NOT NULL,
  Status CHAR(1) DEFAULT 'P' NOT NULL,
    -- Appeal_ID PRIMARY KEY
    CONSTRAINT appeals_crime_id_pk PRIMARY KEY (Appeal_ID),
    -- Crime_ID FOREIGN KEY
    CONSTRAINT appeals_crime_id_fk FOREIGN KEY (Crime_ID)
      REFERENCES Crimes(Crime_ID),
    -- Hearing_date CHECK
    CONSTRAINT appeals_hearing_date_ck CHECK (Filing_date < Hearing_date),
    -- Status CHECK
    CONSTRAINT appeals_status_ck CHECK(Status = 'P'
      OR Status = 'A' OR Status = 'D'));

-------------------------------------------------------------

-- Crime_officers recreation

CREATE TABLE Crime_officers
  (Crime_ID NUMBER(9, 0) NOT NULL,
  Officer_ID NUMBER(8, 0) NOT NULL,
    -- Crime_ID FOREIGN KEY
    CONSTRAINT crime_officers_crime_id_fk FOREIGN KEY (Crime_ID)
      REFERENCES Crimes(Crime_ID),
    -- Officer_ID FOREIGN KEY
    CONSTRAINT crime_officers_officer_id_fk FOREIGN KEY (Officer_ID)
      REFERENCES Officers(Officer_ID));

-------------------------------------------------------------
```

```
-- Crime_charges recreation

CREATE TABLE Crime_charges
  (Charge_ID NUMBER(10, 0),
  Crime_ID NUMBER(9, 0) NOT NULL,
  Crime_code NUMBER(3, 0) NOT NULL,
  Charge_status CHAR(2) DEFAULT 'PD' NOT NULL,
  Fine_amount NUMBER(7, 2) NOT NULL,
  Court_fee NUMBER(7, 2) NOT NULL,
  Amount_paid NUMBER(7, 2) DEFAULT 0 NOT NULL,
  Pay_due_date DATE NOT NULL,
   -- Charge_ID PRIMARY KEY
   CONSTRAINT crime_charges_charge_id_pk PRIMARY KEY (Charge_ID),
   -- Crime_ID FOREIGN KEY
   CONSTRAINT crime_charges_crime_id_fk FOREIGN KEY (Crime_ID)
     REFERENCES Crimes(Crime_ID),
   -- Crime_code FOREIGN KEY
   CONSTRAINT crime_charges_crime_code_fk FOREIGN KEY (Crime_code)
     REFERENCES Crime_codes(Crime_code),
   -- Charge_status CHECK
   CONSTRAINT crime_charges_charge_status_ck CHECK (Charge_status = 'PD'
     OR Charge_status = 'GL' OR Charge_status = 'HG'),
   -- Fine_amount CHECK
   CONSTRAINT crime_charges_fine_amount_ck CHECK (Fine_amount >= 0),
   -- Court_fee CHECK
   CONSTRAINT crime_charges_court_fee_ck CHECK (Court_fee >= 0),
   -- Amount_paid CHECK
   CONSTRAINT crime_charges_amount_paid CHECK (Amount_paid >= 0
```

```
    AND Amount_paid <= (Court_fee + Fine_amount)));

------------------------------------------------------------

-- For testing purposes ...

DESC Criminals;
DESC Aliases;
DESC Crimes;
DESC Sentences;
DESC Prob_officers;
DESC Crime_charges;
DESC Crime_officers;
DESC Officers;
DESC Appeals;
DESC Crime_codes;
```