CREATE TABLE "DEPARTMENT"

( "DEPT\_ID" VARCHAR2(10) NOT NULL ENABLE,

"DEPARTMENT\_NAME" VARCHAR2(50) NOT NULL ENABLE,

"PHONE\_NUMBER" NUMBER(12,0),

"POST\_CODE" NUMBER(9,2),

"HOUSE\_NUMBER" NUMBER(3,0),

"STREET" VARCHAR2(50),

"BUILDING\_NAME" VARCHAR2(100),

PRIMARY KEY ("DEPT\_ID") ENABLE

) ;

CREATE TABLE "REGION"

( "REGION\_ID" VARCHAR2(10),

"REGION\_NAME" VARCHAR2(50) NOT NULL ENABLE,

"COUNTRY\_NAME" VARCHAR2(50) NOT NULL ENABLE,

"TOTAL\_NUMBERS\_PER\_REGION" NUMBER(10,0),

PRIMARY KEY ("REGION\_ID") ENABLE

) ;

CREATE TABLE "CITY"

( "CITY\_ID" VARCHAR2(10) NOT NULL ENABLE,

"CITY\_NAME" VARCHAR2(50) NOT NULL ENABLE,

"LOCATION\_NAME" VARCHAR2(50),

"LOCATION\_ID" VARCHAR2(10) NOT NULL ENABLE,

"POSTCODE" NUMBER(9,2),

"POPULATION" NUMBER(10,0),

"REGION\_ID" VARCHAR2(10),

"HOUSENO" VARCHAR2(10),

"STREET" VARCHAR2(100),

PRIMARY KEY ("CITY\_ID") ENABLE

) ;ALTER TABLE "CITY" ADD CONSTRAINT "FK\_CY" FOREIGN KEY ("LOCATION\_ID")

REFERENCES "CITY" ("CITY\_ID") ENABLE;ALTER TABLE "CITY" ADD FOREIGN KEY ("REGION\_ID")

REFERENCES "REGION" ("REGION\_ID") ENABLE;

**Department Table**

**Region Table**

**City Table**

CREATE TABLE "OFFICER"

( "STAFF\_ID" VARCHAR2(10),

"NAME" VARCHAR2(50),

"RANK" VARCHAR2(20),

"DEPARTMENT" VARCHAR2(50),

"PLACE\_OF\_WORK" VARCHAR2(50),

"HIRE\_DATE" DATE,

"INSPECTOR\_ID" VARCHAR2(10) NOT NULL ENABLE,

"DEPARTMENT\_ID" VARCHAR2(10) NOT NULL ENABLE,

"CITY\_ID" VARCHAR2(10) NOT NULL ENABLE,

PRIMARY KEY ("STAFF\_ID") ENABLE

) ;ALTER TABLE "OFFICER" ADD FOREIGN KEY ("INSPECTOR\_ID")

REFERENCES "OFFICER" ("STAFF\_ID") ENABLE;ALTER TABLE "OFFICER" ADD FOREIGN KEY ("DEPARTMENT\_ID")

REFERENCES "DEPARTMENT" ("DEPT\_ID") ENABLE;ALTER TABLE "OFFICER" ADD FOREIGN KEY ("CITY\_ID")

REFERENCES "CITY" ("CITY\_ID") ENABLE;

CREATE TABLE "OFFENDER"

( "OFFENDER\_ID" VARCHAR2(10),

"NAME" VARCHAR2(50),

"DOB" DATE,

"HOUSE\_NUMBER" NUMBER(3,0),

"STREET" VARCHAR2(20),

"TOWN" VARCHAR2(20),

"COUNTRY" VARCHAR2(20),

"POSTCODE" NUMBER(10,0),

"OCCUPATION" VARCHAR2(20),

"NATIONALITY" VARCHAR2(20),

"RACE" VARCHAR2(20),

"RELIGION" VARCHAR2(20),

PRIMARY KEY ("OFFENDER\_ID") ENABLE

)

CREATE TABLE "CRIME\_COMMITTED"

( "CRIME\_NUMBER" VARCHAR2(10),

"OFFENDER\_ID" VARCHAR2(10),

CONSTRAINT "PK\_CC" PRIMARY KEY ("CRIME\_NUMBER", "OFFENDER\_ID") ENABLE

) ;ALTER TABLE "CRIME\_COMMITTED" ADD FOREIGN KEY ("CRIME\_NUMBER")

REFERENCES "CRIME" ("CRIME\_NUMBER") ENABLE;ALTER TABLE "CRIME\_COMMITTED" ADD FOREIGN KEY ("OFFENDER\_ID")

REFERENCES "OFFENDER" ("OFFENDER\_ID") ENABLE;

**Officer Table**

**Offender Table**

**Crime Committed Table**

insert into region values('NW01', 'North West', 'Greater Manchester', 0);

insert into region values('LN01', 'London', 'Middlesex', 0);

insert into region values('YH01', 'Yorkshire and the Humber', 'West Yorkshire', 0);

insert into region values('SE01', 'South East', 'Cambridgeshire', 0);

insert into region values('EM01', 'East Midlands', 'Nottingshire', 0);

insert into region values('WM01', 'West Midlands', 'West Midlands', 0);

insert into region values('SW01', 'South West', 'Worcestershire', 0);

insert into region values('EE01', 'East of England', 'Bedfordshire', 0);

insert into region values('NE01', 'North East', 'Northumberland', 0);

insert into region values('WAL01', 'Wales', 'Dyfed-Powys', 0);

insert into department (dept\_id, department\_name, phone\_number, post\_code, street, building\_name)

values('GMP001', 'Greater Manchester Police', '0161 856 3221', 'M60 2LA', 'Manchester City Centre Public Enquiry Counter', 'Town Hall')

insert into department values('COL01', 'City of London Police', 02076012220, 'EC2M 4NP', 182, 'Bishopsgate', 'Bishopsgate police station')

insert into department (dept\_id, department\_name, phone\_number, street, building\_name)

values('COL02', 'City of London Police', '999,101', 'PO Box 36451', 'London Police ​Headquarters')

insert into department (dept\_id, department\_name, phone\_number, post\_code, street, building\_name)

values('WYP01', 'West Yorkshire Police', '101', 'LS7 3QU', 'Stainbeck Lane', 'Inner North East Station')

insert into department (DEPT\_ID, DEPARTMENT\_NAME, PHONE\_NUMBER, POST\_CODE, STREET, BUILDING\_NAME)

values('CCP01', 'Cardiff Central Police', '999,101', 'CF10 3NN', 'King Edward VII Avenue', 'Cardiff Central Police Station')

insert into department (DEPT\_ID, DEPARTMENT\_NAME, PHONE\_NUMBER, POST\_CODE, STREET, BUILDING\_NAME)

values('GP01', 'Gwent Police', '01495233956', 'NP23 6UY', 'Bethcar Street, Ebbw Vale', 'Ebbw Vale Police Station')

insert into department (DEPT\_ID, DEPARTMENT\_NAME, PHONE\_NUMBER, POST\_CODE, STREET, BUILDING\_NAME)

values('SWP01', 'South Wales Police', '01656869366', 'CF31 3SU', 'Ty Morgannwg, Bridgend', 'Police Headquarters')

insert into department (DEPT\_ID, DEPARTMENT\_NAME, PHONE\_NUMBER, POST\_CODE, STREET, BUILDING\_NAME)

values('MP01', 'Merseyside Police', '0151 709 6010', 'L1 8JX', 'Canning Place Liverpool', 'Merseyside Police Station')

insert into department (DEPT\_ID, DEPARTMENT\_NAME, PHONE\_NUMBER, POST\_CODE, STREET, BUILDING\_NAME)

values('CC01', 'Cumbria Constabulary', '999,101', 'CA1 3NQ', 'Durranhill Carlisle', 'North Cumbria Headquarters')

insert into department (DEPT\_ID, DEPARTMENT\_NAME, PHONE\_NUMBER, POST\_CODE, STREET, BUILDING\_NAME)

values('DYP01', 'Dyfed-Powys Police Department', '999,101', 'LD7 1HU', 'Ffrydd Newydd, Knighton, Powys', 'Knighton Police Station')

**INSERT INTO REGION**

**INSERT INTO DEPARTMENT**

insert into officer values('001', 'Adrian Leppard', 'Commissioner', 'City of London Police', 'Police Headquarters', '08/13/1984', '001', 'COL01', 'LE01')

insert into officer values('002', 'Ian Dyson', 'Assist. Commissioner', 'City of London', 'Police Headquarters', '07/01/1983', '001', 'COL01', 'Mc001')

insert into officer values('003', 'Wayne Chance', 'Commander of Op.', 'City of London', 'Police Headquarters', '08/13/1986', '002', 'COL01', 'WM01')

insert into officer values('004', 'Anna Ross', 'Sergeant', 'Manchester City Police Department', 'Greater Manchester Police', '03/11/2010', '003', 'GMP001', 'Mc001')

insert into officer values('005', 'Stuart Eckley', 'Inspector', 'Dyfed-Powys Police Department', 'Llandrindod Wells', '12/01/1989', '005', 'DYP01', 'KN01')

insert into officer values('006', 'Jerry Graham', 'Chief Constable', 'Cumbria Constabulary', 'Whitehaven, Cumbria', '01/01/1985', '001', 'CC01', 'EG02')

insert into officer values('007', 'Kevin Warren', 'Inspector', 'Gwent Police', 'Alway, Lliswerry and Ringland', '01/01/1989', '007', 'GMP001', 'Ebb01')

insert into officer values('008', 'Jon Murphy', 'Chief Constable', 'Merseyside Police force', 'Everton, Anfield', '03/30/1975', '008', 'MP01', 'LE01')

insert into officer values('009', 'Matt Scrase ', 'Inspector', 'Dyfed-Powys Police Department', 'Brecon Town', '01/01/1994', '009', 'DYP01', 'KN01')

insert into officer values('010', 'Huw Jones', 'Inspector', 'Gwent Police Department', 'Wyesham, Monmouth', '01/01/1999', '007', 'GMP001', 'Ebb01')

desc city

SELECT \* FROM CITY

DELETE FROM CITY where city\_id='MN01'

alter table city drop column location\_ID cascade constraint

alter table city modify postcode varchar2(10)

insert into city (city\_id, city\_name, location\_name, postcode, population, region\_id, street)

values('Mc001', 'Manchester', 'Old Trafford', 'M16 0RA', 39906, 'NW01', 'Sir Matt Busby Way')

insert into city (city\_id, city\_name, location\_name, postcode, population, region\_id, street)

values('LE01', 'Leeds', 'Leeds Metropolitan University', 'LS6 3QS', '751500', 'YH01', 'The Grange, Headingley Campus')

insert into city (city\_id, city\_name, location\_name, postcode, population, region\_id, street)

values('CAR001', 'Cardiff', 'Cardiff Metropolitan University', 'CF5 2SG', '324800', 'Wal02', 'Western Avenue')

insert into city (city\_id, city\_name, location\_name, postcode, population, region\_id, street)

values('CAR002', 'Cardiff', 'Cardiff University', 'CF10 3AT', '324800', 'Wal02', 'Main Building, Park Place')

insert into city (city\_id, city\_name, location\_name, postcode, population, region\_id, street)

values('WM01', 'Wembly', 'Wembley Football Stadium', 'HA9 0WS', '115079', 'LN01', 'On or near High Road')

insert into city values('EG01', 'Egremont', 'Lowes Court Gallery', 'CA22 2DW', '8194', 'NW01', 12, 'Main Street')

insert into city (city\_id, city\_name, location\_name, postcode, population, region\_id, street)

values('CC01', 'Cardiff City', 'Ninian Park', 'CF11 8AZ', '324800', 'Wal02', 'Leckwith Rd')

insert into city (city\_id, city\_name, location\_name, postcode, population, region\_id, street)

values('Ebb01', 'Ebbw Vale', 'Victoria Business Park', 'NP23 8AN', '33000', 'Wal01', 'Waunllwyd')

insert into city (city\_id, city\_name, location\_name, population, region\_id, houseno, street)

values('MM01', 'Monmouth', 'Cedar Avenue', '91300', 'Wal03', '400', 'West Long Branch')

insert into city values('EG02', 'Egremont', 'Florence Mine', 'CA22 2DB', 8194, 'NW01', 70, 'Main Street')

insert into city (city\_id, city\_name, location\_name, population, region\_id, street)

values('KN01', 'Knighton', 'Milebrook House Hotel', '4099', 'WAL01', 'Knighton, Powys')

**INSERT INTO OFFICER**

**INSERT INTO CITY**

//CRIME TABLE

alter table crime modify crime\_date date

select sysdate from dual

DESC CRIME

SELECT \* FROM CRIME

alter table crime modify crime\_desc varchar2(200)

alter table crime add constraint fk\_rc foreign key (related\_crimes) referencing crime(crime\_number)

insert into crime (crime\_number, crime\_desc, crime\_date, crime\_notes, crime\_status, crime\_type, outside\_source\_of\_report\_desc, date\_closed, staff\_id, location\_id)

values('HOM001', 'Murder', '08/15/2014', 'Suspect murdered victim using blunt object', 'CLOSED', 'Homicide', 'CCTV Footage, murder weapon', '05/30/2014','003', 'WM01');

insert into crime (crime\_number, crime\_desc, crime\_date, crime\_notes, crime\_status, crime\_type, outside\_source\_of\_report\_desc, staff\_id, location\_id)

values('ROB01', 'Bank Robbery', '09/15/2014', 'Suspects are still at large', 'OPEN', 'Robbery', 'CCTV Footage, Witness testimonials', '002', 'Mc001');

insert into crime (crime\_number, crime\_desc, crime\_date, crime\_notes, crime\_status, crime\_type, outside\_source\_of\_report\_desc, date\_closed, staff\_id, location\_id)

values('VC001', 'Sever Wounding', '06/10/2014', 'Suspect inflicted severe injuries on victim, immediate medical attention required.', 'CLOSED', 'Violent Crime', 'Witness statement', '06/30/2014','001', 'LE01');

insert into crime (crime\_number, crime\_desc, crime\_date, crime\_notes, crime\_status, crime\_type, outside\_source\_of\_report\_desc, staff\_id, location\_id)

values('VC002', 'Domestic Violence', '09/15/2014', 'Crime involving partners', 'OPEN', 'Violent Crime', 'Partner Witness testimonials', '004', 'Mc001');

insert into crime (crime\_number, crime\_desc, crime\_date, crime\_notes, crime\_status, crime\_type, outside\_source\_of\_report\_desc, staff\_id, location\_id)

values('ROB02', 'Business Robbery', '10/01/2014', 'Suspect robbed a shop with the aid of an accomplice', 'OPEN', 'Robbery', 'CCTV Footage', '004', 'LE01');

insert into crime (crime\_number, crime\_desc, crime\_date, crime\_notes, crime\_status, crime\_type, outside\_source\_of\_report\_desc, staff\_id, location\_id)

values('VC003', 'Domestic Violence', '11/20/2014', 'Suspect hit victim with a blunt object, minor injury recorded.', 'CLOSED', 'Violent Crime', 'Victim''s Testimony', '005', 'KN01');

insert into crime (crime\_number, crime\_desc, crime\_date, crime\_notes, crime\_status, crime\_type, outside\_source\_of\_report\_desc, related\_crimes, staff\_id, location\_id)

values('BURG01', 'Domestic Burglary', '12/25/2014', 'Suspect broke into victim''s apartment, several items were reportedly missing.', 'OPEN', 'Theft', 'CCTV footage, fingerprints.', 'VC003', '006', 'EG02');

insert into crime (crime\_number, crime\_desc, crime\_date, crime\_notes, crime\_status, crime\_type, outside\_source\_of\_report\_desc, related\_crimes, staff\_id, location\_id)

values('PPT01', 'Personal property theft', '04/22/2014', 'Suspect stole item of value from victim.', 'CLOSED', 'Theft', 'Eye witness statement.', 'VC001', '001', 'LE01');

insert into crime (crime\_number, crime\_desc, crime\_date, crime\_notes, crime\_status, crime\_type, outside\_source\_of\_report\_desc, staff\_id, location\_id)

values('TV01', 'Theft of vehicle', '06/10/2014', 'Suspect drove victim''s vehicle away illegally.', 'OPEN', 'Theft', 'Eye witness statement, CCTV footage.', '008', 'Mc001');

insert into crime (crime\_number, crime\_desc, crime\_date, crime\_notes, crime\_status, crime\_type, outside\_source\_of\_report\_desc, related\_crimes, staff\_id, location\_id)

values('IF01', 'Internet Fraud', '12/10/2014', 'Suspect defrauded victim of large sum of money', 'OPEN', 'Fraud', 'Data from National Fraud Intelligence Bureau(NFIB)', 'HOM001', '008', 'Ebb01');

**INSERT INTO CRIME**

//OFFENDER TABLE

select \* from offender

desc offender

alter table offender modify postcode varchar2(10)

//INSERTED

insert into offender values('OFF001', 'John Doe', '01/23/1975', 224, 'Hawthorne road', 'Bootle', 'Merseyside', 'L20', 'Student', 'Malaysia', 'White', 'Islam')

insert into offender values('OFF002', 'James Cornor', '12/13/1986', 12, 'Parsoloes Patrk road', 'Barking', 'Greater London', 'IG11', 'Technician', 'Indian', 'Asian', 'Hindu')

insert into offender values('OFF003', 'Evans Stone', '10/19/1979', 151, 'Ferry Rd', 'Isle Of Dogs', 'East London', 'E14 3BT', 'Electrician', 'England', 'White', 'Christian')

insert into offender values('OFF004', 'Ian Pupa', '02/09/1990', 98, 'High School Rd.', 'Earl Shilton', 'Leicestershire', 'LE9', 'Musician', 'Pakistan', 'Asian', 'Islam')

insert into offender values('OFF005', 'Van Stein', '10/27/1988', 70, 'Longelandes way', 'Banbury', 'Oxfordshire', 'OX16', 'Technician', 'Indian', 'Asian', 'Hindu')

insert into offender values('OFF006', 'Joe Forg', '09/15/1987', 14, 'Wilson Rd', 'Ely', 'Cardiff', 'CF5', 'Footballer', 'Scotland', 'White', 'Others')

insert into offender values('OFF007', 'Hummy Kent', '12/20/1981', 17, 'Abergavenny', 'Ebbw Vale', 'Gwent', 'IG11', 'Trader', 'Scotland', 'White', 'Islam')

insert into offender values('OFF008', 'Jeremy Flint', '12/25/1984', 11, 'Old Street', 'Hoxton', 'Central London', 'N1', 'Entertainer', 'Nigerian', 'Black', 'Islam')

insert into offender values('OFF009', 'David Jones', '06/23/1988', 03, 'Stadium Road', 'Wembley', 'Middlesex', 'HA0', 'Footballer', 'Nigerian', 'Black', 'Christian')

insert into offender values('OFF010', 'Jack Jones', '08/23/1991', 15, 'Main Street', 'Manchester', 'Greater Manchester', 'M60', 'Footballer', 'Mozambique', 'Black', 'Islam')

//CRIMES COMMITTED

DESC CRIME\_COMMITTED

INSERT INTO CRIME\_COMMITTED VALUES ('ROB01', 'OFF001')

INSERT INTO CRIME\_COMMITTED VALUES ('VC001', 'OFF005')

INSERT INTO CRIME\_COMMITTED VALUES ('TV01', 'OFF008')

INSERT INTO CRIME\_COMMITTED VALUES ('VC002', 'OFF010')

INSERT INTO CRIME\_COMMITTED VALUES ('VC003', 'OFF007')

INSERT INTO CRIME\_COMMITTED VALUES ('IF01', 'OFF006')

INSERT INTO CRIME\_COMMITTED VALUES ('HOM001', 'OFF003')

INSERT INTO CRIME\_COMMITTED VALUES ('ROB02', 'OFF002')

INSERT INTO CRIME\_COMMITTED VALUES ('BURG01', 'OFF004')

INSERT INTO CRIME\_COMMITTED VALUES ('PPT01', 'OFF009')

SELECT \* FROM REGION

SELECT \* FROM OFFENDER

SELECT \* FROM CRIME

SELECT \* FROM CITY

SELECT \* FROM CRIME WHERE LOCATION\_ID = 'Ebb01'

UPDATE REGION SET TOTAL\_NUMBER\_PER\_REGION = 4 WHERE REGION\_ID = 'NW01'

UPDATE REGION SET TOTAL\_NUMBER\_PER\_REGION = 3 WHERE REGION\_ID = 'YH01'

UPDATE REGION SET TOTAL\_NUMBER\_PER\_REGION = 2 WHERE REGION\_ID = 'Wal01'

UPDATE REGION SET TOTAL\_NUMBER\_PER\_REGION = 1 WHERE REGION\_ID = 'LN01'

**INSERT INTO OFFENDER**

**INSERT INTO CRIME\_COMMITTED**

**SOME DML**

//--- TRIGGERS ---\\

//MAKE A TRIGGER THAT WILL AUTOMATICALLY UPDATE THE REGION TABLE IF NEW CRIMES ARE RECORDED IN A PARTICULAR REGION.

//SINCE THERE IS NO DIRECT RELATIONSHIP BETWEEN CRIME AND REGION TABLES, THE LOCATION\_ID ON THE CRIME TABLE SHOULD BE SPLIT INTO TWO (REGION\_ID & CITY\_ID).

//ALSO, THERE IS NEED TO CHECK THE NUMBER OF CRIMES REPORTED IN A PARTICULAR CITY. SO, A NEW ROW WILL BE ADDED TO THE CITY TABLE AND A TRIGGER WILL BE CREATED TO CALCULATE THIS ROW.

//FIRST ALTER CRIME AND CHANGE LOCATION\_ID TO CITY\_ID

ALTER TABLE CRIME RENAME COLUMN LOCATION\_ID TO CITY\_ID

//CHECK

SELECT \* FROM CRIME

//SECONDLY ADD A NEW COLUMN NAMED REGION\_ID AND REFERENCE IT TO REGION\_ID IN THE REGION TABLE

ALTER TABLE CRIME ADD REGION\_ID VARCHAR2(10) REFERENCES REGION(REGION\_ID)

//CHECK

SELECT \* FROM CRIME

//NOW INSERT VALUES INTO THE NEWLY CREATED COLUMN IN CRIME TABLE

UPDATE CRIME SET REGION\_ID = 'NW01' WHERE CITY\_ID = 'Mc001'

UPDATE CRIME SET REGION\_ID = 'YH01' WHERE CITY\_ID = 'LE01'

UPDATE CRIME SET REGION\_ID = 'WAL01' WHERE CITY\_ID = 'KN01'

UPDATE CRIME SET REGION\_ID = 'WAL01' WHERE CITY\_ID = 'Ebb01'

UPDATE CRIME SET REGION\_ID = 'LN01' WHERE CITY\_ID = 'WM01'

UPDATE CRIME SET REGION\_ID = 'NW01' WHERE CITY\_ID = 'EG02'

//CHECK

SELECT \* FROM CRIME

//NEXT ADD A NEW COLUMN IN CITY TABLE THAT STORES THE TOTAL CRIME IN EACH CITY.

ALTER TABLE CITY ADD TOTAL\_CRIME\_PER\_CITY NUMBER(11)

//CHECK

SELECT \* FROM CITY

//INSERT AN INITIAL VALUE OF ZERO TO THE NEWLY CREATED ROW IN CITY TABLE

UPDATE CITY SET TOTAL\_CRIME\_PER\_CITY =0

//CHECK

SELECT \* FROM CITY

//MANUALLY COUNT (OR USE SQL TO COUNT) THE NUMBER OF RECORDED CRIMES IN EACH CITY AND INSERT THE CORRESPONDING VALUE.

SELECT \* FROM CRIME

UPDATE CITY SET TOTAL\_CRIME\_PER\_CITY = 3 WHERE CITY\_ID = 'Mc001'

UPDATE CITY SET TOTAL\_CRIME\_PER\_CITY = 3 WHERE CITY\_ID = 'LE01'

UPDATE CITY SET TOTAL\_CRIME\_PER\_CITY = 1 WHERE CITY\_ID = 'KN01'

**TRIGGERS**

UPDATE CITY SET TOTAL\_CRIME\_PER\_CITY = 1 WHERE CITY\_ID = 'Ebb01'

UPDATE CITY SET TOTAL\_CRIME\_PER\_CITY = 1 WHERE CITY\_ID = 'WM01'

UPDATE CITY SET TOTAL\_CRIME\_PER\_CITY = 1 WHERE CITY\_ID = 'EG02'

//CHECK

SELECT \* FROM CITY

SELECT \* FROM REGION

//NOW CREATE A TRIGGER THAT WILL UPDATE THE TOTAL\_CRIME\_PER\_CITY COLUMN IN THE CITY TABLE WHENEVER A NEW CRIME IS RECORDED.

CREATE OR REPLACE TRIGGER UPDATE\_CITY

AFTER INSERT OR UPDATE OF CITY\_ID

ON CRIME

FOR EACH ROW

BEGIN

UPDATE CITY SET TOTAL\_CRIME\_PER\_CITY = TOTAL\_CRIME\_PER\_CITY + 1

WHERE CITY\_ID = :NEW.CITY\_ID;

END;

//CHECK

DESC REGION

DESC OFFICER

DESC CRIME

//AND FINALLY CREATE A TRIGGER THAT WILL UPDATE THE TOTAL\_NUMBER\_PER\_REGION ON THE REGION TABLE

CREATE OR REPLACE TRIGGER UPDATE\_REGION

AFTER INSERT OR UPDATE OF REGION\_ID

ON CRIME

FOR EACH ROW

BEGIN

UPDATE REGION SET TOTAL\_NUMBER\_PER\_REGION = TOTAL\_NUMBER\_PER\_REGION +1

WHERE REGION\_ID =:NEW.REGION\_ID;

END;

**TRIGGERS (CONTINUED)**

**DATABASE SYSTEM REQUIREMENTS (CASE SCENARIO)**

//DATABASE SYSTEM REQUIREMENTS\\

1.) Ability to record and retrieve crime, date and crime code

select crime\_number as "Crime Code", crime\_desc, crime\_date from crime

2.) Ability to identify the police officer who recorded notes about the crime and also details of the person who reported the crime

select o.name as "Officer Name", c.crime\_desc, c.crime\_notes from officer o, crime c where c.staff\_id = o.staff\_id

3.) Ability to retrieve crime data for each county and city

SELECT COUNTY\_NAME, TOTAL\_NUMBER\_PER\_REGION FROM REGION WHERE TOTAL\_NUMBER\_PER\_REGION >0

OR

SELECT C.CRIME\_DESC, R.COUNTY\_NAME, R.TOTAL\_NUMBER\_PER\_REGION FROM CRIME C, REGION R

WHERE TOTAL\_NUMBER\_PER\_REGION>0 AND

C.REGION\_ID = R.REGION\_ID

4.) Police would like to retrieve data on closed and open crimes

select crime\_number, crime\_desc, crime\_status, crime\_date, date\_closed from crime

5.) Ability to record data about the offender and the objects used to commit the crime

SELECT O.NAME AS "OFFENDER NAME", C.CRIME\_NOTES, C.CRIME\_DESC FROM OFFENDER O, CRIME C

6.) Ability to identify if a crime is associated with one or many events

SELECT CRIME\_NUMBER, CRIME\_DESC, RELATED\_CRIMES FROM CRIME

WHERE RELATED\_CRIMES !='NULL'

7.)//NO WITNESS AND VICTIM TABLES\\

8.) Ability to record and retrieve details about the offender's history if he has committed more than one crime

//BEFORE THIS CAN BE DONE SUCCESSFULLY, CERTAIN ALTERATIONS HAS TO BE CARRIED OUT ON THE TABLES.

//THERE SHOULD BE A MEANS TO IDENTIFY WHICH OFFENDER COMMITTED A PARTICULAR CRIME. FOR THIS REASON, THE CRIME TABLE SHOULD BE ALTERED TO INCLUDE OFFENDER ID

alter table crime add offender\_id varchar2(10)

alter table crime add constraint fk\_oid foreign key (offender\_id) references offender (offender\_id)

//NOW A NEW TRIGGER SHOULD BE CREATED TO AUTOMATICALLY UPDATE THE CRIME\_COMMITTED TABLE WHENEVER A NEW CRIME IS RECORDED

//FIRST UPDATE THE CRIME TABLE WITH THE OFFENDERS

UPDATE CRIME SET OFFENDER\_ID = 'OFF004' WHERE CRIME\_NUMBER = 'BURG01'

UPDATE CRIME SET OFFENDER\_ID = 'OFF002' WHERE CRIME\_NUMBER = 'ROB02'

UPDATE CRIME SET OFFENDER\_ID = 'OFF003' WHERE CRIME\_NUMBER = 'HOM001'

UPDATE CRIME SET OFFENDER\_ID = 'OFF005' WHERE CRIME\_NUMBER = 'VC001'

UPDATE CRIME SET OFFENDER\_ID = 'OFF006' WHERE CRIME\_NUMBER = 'IF01'

UPDATE CRIME SET OFFENDER\_ID = 'OFF007' WHERE CRIME\_NUMBER = 'VC003'

UPDATE CRIME SET OFFENDER\_ID = 'OFF008' WHERE CRIME\_NUMBER = 'TV01'

UPDATE CRIME SET OFFENDER\_ID = 'OFF009' WHERE CRIME\_NUMBER = 'PPT01'

UPDATE CRIME SET OFFENDER\_ID = 'OFF010' WHERE CRIME\_NUMBER = 'VC002'

UPDATE CRIME SET OFFENDER\_ID = 'OFF001' WHERE CRIME\_NUMBER = 'ROB01'

//CHECK

SELECT \* FROM CRIME

//FINALLY, THE TRIGGER

create or replace trigger ins\_cc

after insert or update of crime\_number

on crime

for each row

begin

insert into crime\_committed (crime\_number)

values(:New.crime\_number);

end;

create or replace trigger ins\_cc\_offend

after insert or update of offender\_id

on crime

for each row

begin

insert into crime\_committed (offender\_id)

values(:New.offender\_id);

//FINALLY

select o.\*, c.related\_crimes from offender o, crime c

where related\_crimes !='NULL'

and c.offender\_id = o.offender\_id

9.) If the crime is solved the police is able to retrieve information about the date it was solved, the detective who solved it and the who was the criminal.

If it's still pending, who is the detective who investigates it and the department they are working in.

//FOR SOLVED CRIME

SELECT C.CRIME\_DESC, C.CRIME\_DATE, C.CRIME\_STATUS, C.DATE\_CLOSED, F.NAME AS "CRIMINAL", O.NAME AS "DETECTIVE-IN-CHARGE" FROM CRIME C, OFFENDER F, OFFICER O

WHERE CRIME\_STATUS = 'CLOSED'

AND O.STAFF\_ID = C.STAFF\_ID

AND C.OFFENDER\_ID = F.OFFENDER\_ID

9.) \\FOR PENDING CRIMES

SELECT C.CRIME\_DESC, C.CRIME\_DATE, C.CRIME\_STATUS, O.NAME AS "DETECTIVE-IN-CHARGE", D.DEPARTMENT\_NAME FROM CRIME C, OFFICER O, DEPARTMENT D

WHERE CRIME\_STATUS = 'OPEN'

AND C.STAFF\_ID = O.STAFF\_ID

AND O.DEPARTMENT\_ID = D.DEPT\_ID

//

10. and 11. are recursive relationships. Need to figure out how to handle that

\\

10.) Every police officer that recorded notes for a crime has to report to an inspector.

SELECT NAME, RANK, DEPARTMENT FROM OFFICER WHERE INSPECTOR\_ID LIKE '%0%'

11.) Every crime has at least one suspect and the police can see if any suspect is previously convicted or accused of another crime

12.) Show details of offenses that have the same officer who supervised an offense happened the 19th of December, 2014.

SELECT \* FROM CRIME WHERE CRIME\_DATE = '09/15/2014'

13.) list which crimes are related to which, and make it clear by displaying the name, date and number of each crime

14.) We need to know the crime details, along with the details of the officer who reported it

SELECT C.CRIME\_NUMBER, C.CRIME\_DESC, C.CRIME\_DATE, C.CRIME\_STATUS, O.NAME FROM CRIME C, OFFICER O

WHERE C.STAFF\_ID = O.STAFF\_ID

15.) Which department has the highest number of open crimes

SELECT DEPARTMENT\_NAME FROM DEPARTMENT WHERE DEPT\_ID IN (SELECT DEPT\_ID FROM OFFICER WHERE STAFF\_ID IN (SELECT STAFF\_ID FROM CRIME WHERE CRIME\_STATUS = 'OPEN'))

SELECT \* FROM CRIME

SELECT \* FROM DEPARTMENT

SELECT \* FROM OFFICER

//

DEADLOCK!!!

\\

SELECT DEPARTMENT\_NAME FROM CRIME WHERE CRIME\_STATUS = 'OPEN' AND DEPT\_ID IN (SELECT DEPT\_ID FROM DEPARTMENT WHERE )

//

DEADLOCK!!!

\\

16.) LIST OF ALL CLOSED CRIMES WHERE THE SUSPECT IS A SUSPECT MORE THAN THREE TIMES

SELECT CRIME\_DESC FROM CRIME WHERE CRIME\_STATUS = 'CLOSED' AND COUNT()>3

**DATABASE SYSTEM REQUIREMENTS (CASE SCENARIO CONTINUED)**

//CHECK

SELECT \* FROM CRIME

//FINALLY, THE TRIGGER

create or replace trigger ins\_cc

after insert or update of crime\_number

on crime

for each row

begin

insert into crime\_committed (crime\_number)

values(:New.crime\_number);

end;

create or replace trigger ins\_cc\_offend

after insert or update of offender\_id

on crime

for each row

begin

insert into crime\_committed (offender\_id)

values(:New.offender\_id);

//FINALLY

select o.\*, c.related\_crimes from offender o, crime c

where related\_crimes !='NULL'

and c.offender\_id = o.offender\_id

9.) If the crime is solved the police is able to retrieve information about the date it was solved, the detective who solved it and the who was the criminal.

If it's still pending, who is the detective who investigates it and the department they are working in.

//FOR SOLVED CRIME

SELECT C.CRIME\_DESC, C.CRIME\_DATE, C.CRIME\_STATUS, C.DATE\_CLOSED, F.NAME AS "CRIMINAL", O.NAME AS "DETECTIVE-IN-CHARGE" FROM CRIME C, OFFENDER F, OFFICER O

WHERE CRIME\_STATUS = 'CLOSED'

AND O.STAFF\_ID = C.STAFF\_ID

AND C.OFFENDER\_ID = F.OFFENDER\_ID

9.) \\FOR PENDING CRIMES

SELECT C.CRIME\_DESC, C.CRIME\_DATE, C.CRIME\_STATUS, O.NAME AS "DETECTIVE-IN-CHARGE", D.DEPARTMENT\_NAME FROM CRIME C, OFFICER O, DEPARTMENT D

WHERE CRIME\_STATUS = 'OPEN'

AND C.STAFF\_ID = O.STAFF\_ID

AND O.DEPARTMENT\_ID = D.DEPT\_ID

//

10. and 11. are recursive relationships. Need to figure out how to handle that

\\

10.) Every police officer that recorded notes for a crime has to report to an inspector.

SELECT NAME, RANK, DEPARTMENT FROM OFFICER WHERE INSPECTOR\_ID LIKE '%0%'

11.) Every crime has at least one suspect and the police can see if any suspect is previously convicted or accused of another crime

12.) Show details of offenses that have the same officer who supervised an offense happened the 19th of December, 2014.

SELECT \* FROM CRIME WHERE CRIME\_DATE = '09/15/2014'

13.) list which crimes are related to which, and make it clear by displaying the name, date and number of each crime

14.) We need to know the crime details, along with the details of the officer who reported it

SELECT C.CRIME\_NUMBER, C.CRIME\_DESC, C.CRIME\_DATE, C.CRIME\_STATUS, O.NAME FROM CRIME C, OFFICER O

WHERE C.STAFF\_ID = O.STAFF\_ID

15.) Which department has the highest number of open crimes

SELECT DEPARTMENT\_NAME FROM DEPARTMENT WHERE DEPT\_ID IN (SELECT DEPT\_ID FROM OFFICER WHERE STAFF\_ID IN (SELECT STAFF\_ID FROM CRIME WHERE CRIME\_STATUS = 'OPEN'))

SELECT \* FROM CRIME

SELECT \* FROM DEPARTMENT

SELECT \* FROM OFFICER

//

DEADLOCK!!!

\\

SELECT DEPARTMENT\_NAME FROM CRIME WHERE CRIME\_STATUS = 'OPEN' AND DEPT\_ID IN (SELECT DEPT\_ID FROM DEPARTMENT WHERE )

//

DEADLOCK!!!

\\

16.) LIST OF ALL CLOSED CRIMES WHERE THE SUSPECT IS A SUSPECT MORE THAN THREE TIMES

SELECT CRIME\_DESC FROM CRIME WHERE CRIME\_STATUS = 'CLOSED' AND COUNT()>3

**DATABASE SYSTEM REQUIREMENTS (CASE SCENARIO CONTINUED)**

// VIEWS \\

//VIEW 1\\

create view view\_offender

as

select o.name as "Officer Name", c.crime\_desc, c.crime\_notes

from officer o, crime

c where c.staff\_id = o.staff\_id

order by o.name

select \* from view\_offender

//VIEW 2\\

DROP VIEW city\_view

select \* from city\_view

create view city\_view

as

SELECT C.CRIME\_DESC, R.COUNTY\_NAME, R.TOTAL\_NUMBER\_PER\_REGION

FROM CRIME C, REGION R

WHERE TOTAL\_NUMBER\_PER\_REGION>0 AND

C.REGION\_ID = R.REGION\_ID

ORDER BY TOTAL\_NUMBER\_PER\_REGION DESC

//VIEW 4\\

create view solved\_crime\_view

as

SELECT C.CRIME\_DESC, C.CRIME\_DATE, C.CRIME\_STATUS, C.DATE\_CLOSED, F.NAME AS "CRIMINAL", O.NAME AS "DETECTIVE-IN-CHARGE"

FROM CRIME C, OFFENDER F, OFFICER O

WHERE CRIME\_STATUS = 'CLOSED'

AND O.STAFF\_ID = C.STAFF\_ID

AND C.OFFENDER\_ID = F.OFFENDER\_ID

select \* from solved\_crime\_view

//VIEW 5\\

create view pending\_crime\_view

as

SELECT C.CRIME\_DESC, C.CRIME\_DATE, C.CRIME\_STATUS, O.NAME AS "DETECTIVE-IN-CHARGE", D.DEPARTMENT\_NAME FROM CRIME C, OFFICER O, DEPARTMENT D

WHERE CRIME\_STATUS = 'OPEN'

AND C.STAFF\_ID = O.STAFF\_ID

AND O.DEPARTMENT\_ID = D.DEPT\_ID

select \* from pending\_crime\_view

//VIEW 6\\

create view officer\_view

as

SELECT NAME, RANK, DEPARTMENT FROM OFFICER WHERE INSPECTOR\_ID LIKE '%0%'

**VIEWS**

**PROCEDURAL CONSTRAINT TRIGGERS**

//PROCEDURAL CONSTRAINT TRIGGERS\\

1.) CREATE A TRIGGER TO CHECK IF THE CRIME DATE IS VALID. DATES GREATER THAN TODAY'S DATE ARE INVALID

CREATE OR REPLACE TRIGGER CONSTRAINT\_TRIGG1

BEFORE INSERT OR UPDATE OF CRIME\_DATE

ON CRIME

FOR EACH ROW

BEGIN

IF :NEW.CRIME\_DATE > SYSDATE THEN

RAISE\_APPLICATION\_ERROR (-20001, 'CRIME DATE CANNOT BE IN THE FUTURE');

END IF;

END;

SELECT \* FROM CRIME

2.) CREATE A TRIGGER TO CHECK IF THEN FOREIGN EXIST BEFORE INSERTING IT INTO THE CHILD TABLE.

CREATE OR REPLACE TRIGGER CONSTRAINT\_TRIGG2

BEFORE INSERT OR UPDATE OF STAFF\_ID

ON CRIME

FOR EACH ROW

DECLARE

DUMMY NUMBER(11);

BEGIN

SELECT COUNT(\*)

INTO DUMMY

FROM OFFICER

WHERE STAFF\_ID = :NEW.STAFF\_ID;

IF (DUMMY < 1) THEN

RAISE\_APPLICATION\_ERROR(-20001, 'THE STAFF WITH THIS ID: ' ||:NEW.STAFF\_ID ||'DOES NOT EXIST IN THE OFFICER TABLE.');

END IF;

END;

//SEQUENCE\\

CREATE SEQUENCE auto\_id

minvalue 100

maxvalue 999999999

increment by 1

nocache;

CREATE SEQUENCE seq\_id

MINVALUE 1

INCREMENT BY 1

NOCACHE;

**SEQUENCE**

//USER VALIDATION\\

drop table my\_users

create table my\_users

(user\_name varchar2(10),

user\_pwd varchar2(10),

user\_type char(1));

insert into my\_users values('almond', 'monalex', 'A');

insert into my\_users values('monalex', 'monalex','U');

CREATE OR REPLACE FUNCTION

VALIDATE\_USERS\_FROM\_DB (P\_USERNAME IN VARCHAR2(10), P\_PASSWORD IN VARCHAR2(10))

RETURN BOOLEAN

AS V\_PW\_CHECK VARCHAR2(1);

BEGIN

SELECT 'X'

INTO V\_PW\_CHECK

FROM MY\_USERS

WHERE UPPER(USER\_NAME) = UPPER(P\_USERNAME)

AND USER\_PWD = P\_PASSWORD

APEX\_UTIL.SET\_AUTHENTICATION\_RESULT(0);

RETURN TRUE;

EXCEPTION WHEN NO\_DATA\_FOUND THEN

APEX\_UTIL.SET\_AUTHENTICATION\_RESULT(4);

RETURN FALSE;

END VALIDATE\_USERS\_FROM\_DB;

DECLARE

VRESULT VARCHAR2(10);

BEGIN

IF VALIDATE\_USER('almond', 'monalex') THEN

DBMS\_OUTPUT.PUT\_LINE('OK');

ELSE

DBMS\_OUTPUT.PUT\_LINE('ERROR');

END IF;

END;

//CLASSIC REPORT CODES\\

SELECT \* FROM CITY

SELECT CITY\_NAME, POPULATION, TOTAL\_CRIME\_PER\_CITY,

CASE WHEN TOTAL\_CRIME\_PER\_CITY<1 THEN 'GREEN'

WHEN TOTAL\_CRIME\_PER\_CITY BETWEEN 1 AND 2 THEN 'ORANGE'

WHEN TOTAL\_CRIME\_PER\_CITY > 2 THEN 'RED'

END THE\_COLOUR

FROM CITY

//REGION MAP CODES\\

select 'f?p=&APP\_ID.:2:'||:app\_session||'::::::' link, REGION\_NAME label, TOTAL\_NUMBER\_PER\_REGION value1

from "MONDAY ALEXANDER ABEL"."REGION"

**USER VALIDATION**

**APEX**