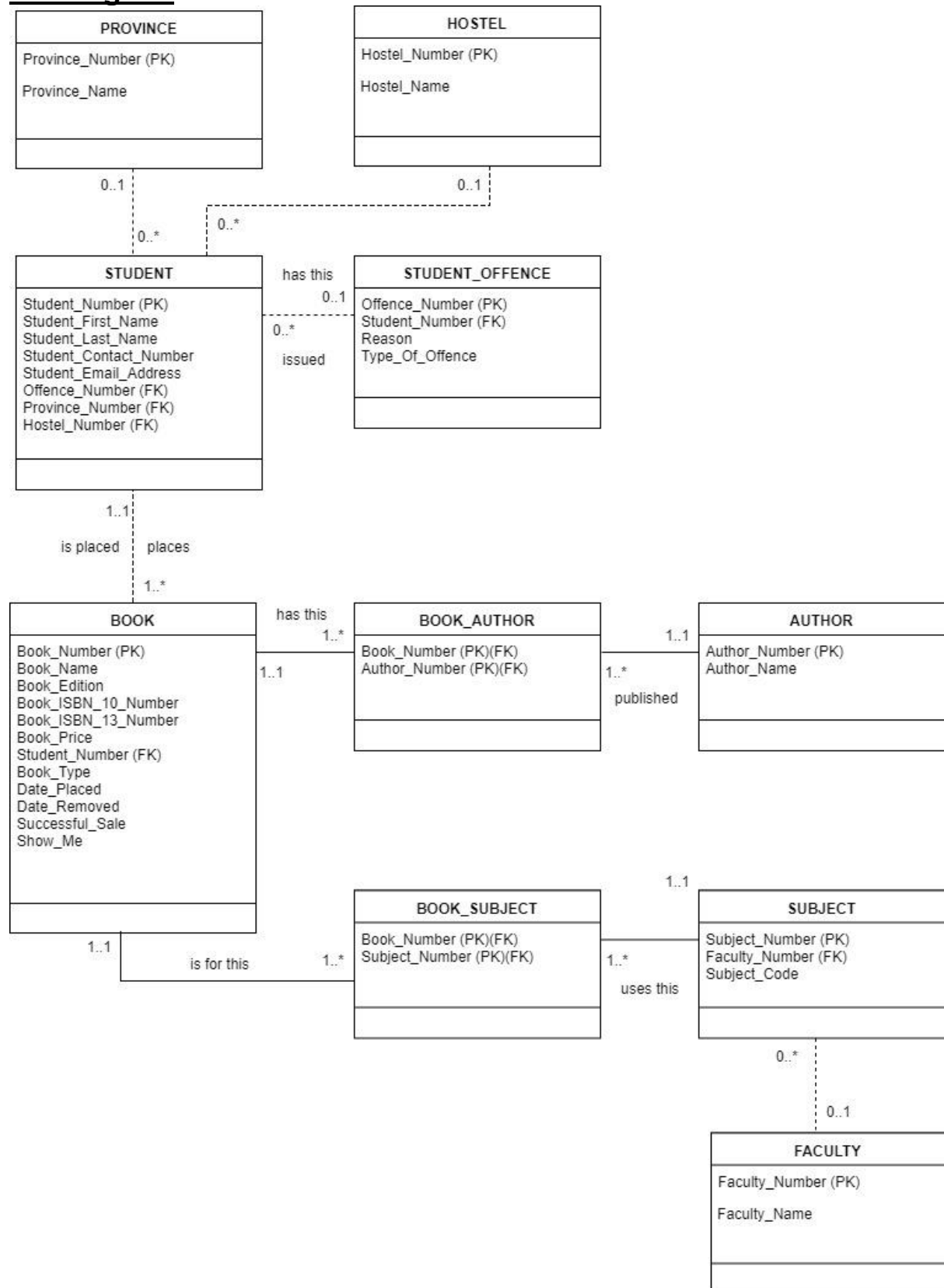


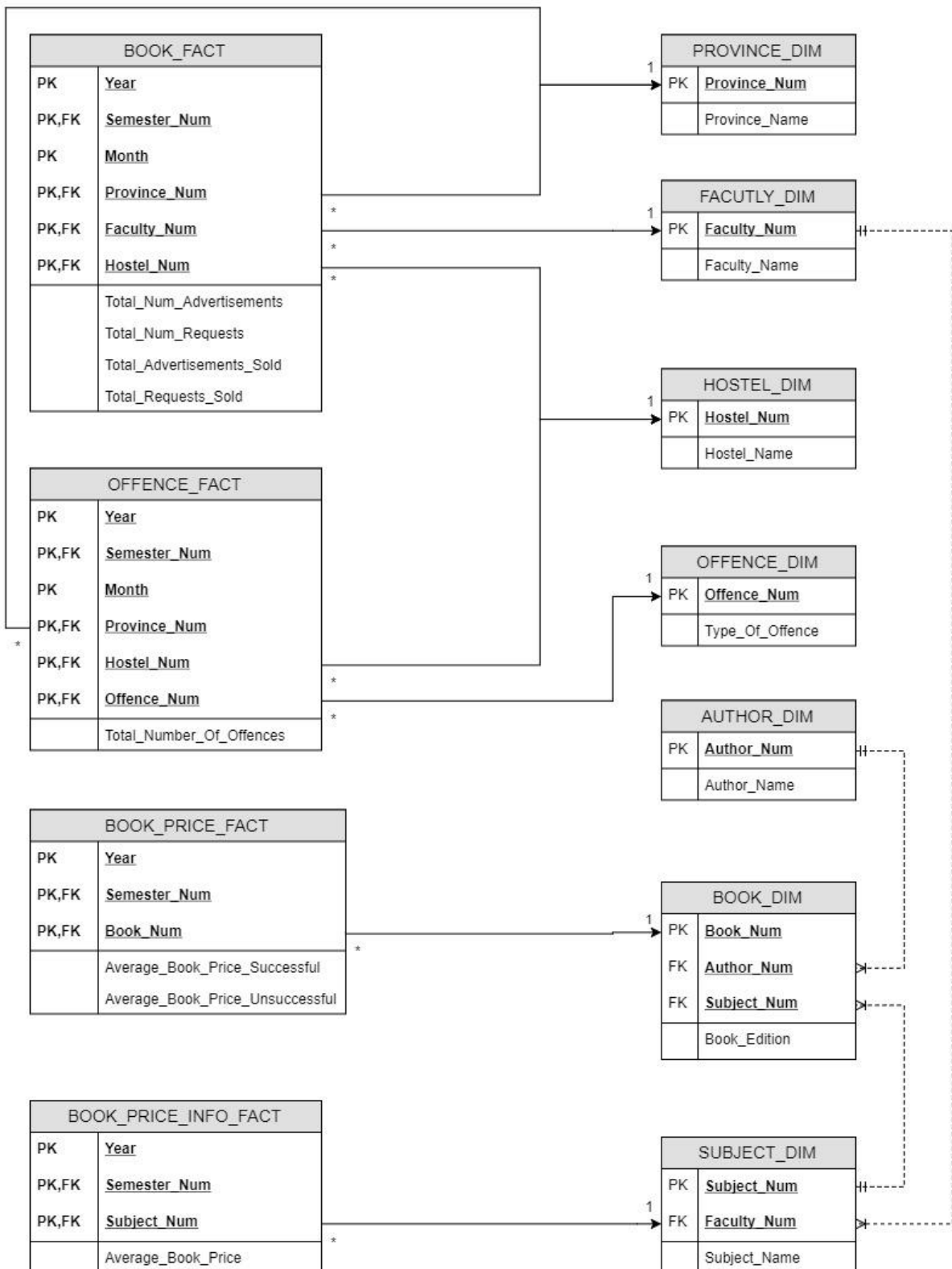
Table of Contents

ER Diagram.....	2
Star Schema.....	3
Creation of DSS Tables	4
Populations of Facts and Dimensions	6
Book Fact.....	6
Book Price Info Fact.....	8
Book Price Fact.....	10
Offence Fact	13
Creation Of Indexes.....	15
User Views	16
Update Of Tables	17

ER Diagram



Star Schema



Creation of DSS Tables

```
/*
DROP TABLE PROVINCE_DIM;
DROP TABLE FACULTY_DIM;
DROP TABLE HOSTEL_DIM;
DROP TABLE SEMESTER_DIM;
DROP TABLE OFFENCE_DIM;
DROP TABLE AUTHOR_DIM;
DROP TABLE SUBJECT_DIM;
DROP TABLE BOOK_DIM;
DROP TABLE BOOK_FACT;
DROP TABLE OFFENCE_FACT;
DROP TABLE BOOK_PRICE_INFO_FACT;
DROP TABLE BOOK_PRICE_FACT;

PURGE TABLE PROVINCE_DIM;
PURGE TABLE FACULTY_DIM;
PURGE TABLE HOSTEL_DIM;
PURGE TABLE SEMESTER_DIM;
PURGE TABLE OFFENCE_DIM;
PURGE TABLE AUTHOR_DIM;
PURGE TABLE SUBJECT_DIM;
PURGE TABLE BOOK_DIM;
PURGE TABLE BOOK_FACT;
PURGE TABLE OFFENCE_FACT;
PURGE TABLE BOOK_PRICE_INFO_FACT;
PURGE TABLE BOOK_PRICE_FACT;
*/

CREATE TABLE PROVINCE_DIM
(
    Province_Number      NUMBER(2)      NOT NULL,
    Province_Name        VARCHAR2(20),
    CONSTRAINT PK_STAR_Province_Number PRIMARY KEY (Province_Number)
);

CREATE TABLE FACULTY_DIM
(
    Faculty_Number       NUMBER(2)      NOT NULL,
    Faculty_Name         VARCHAR2(40),
    CONSTRAINT PK_STAR_Faculty_Number PRIMARY KEY (Faculty_Number)
);

CREATE TABLE HOSTEL_DIM
(
    Hostel_Number        NUMBER(2)      NOT NULL,
    Hostel_Name          VARCHAR2(20),
    CONSTRAINT PK_STAR_Hostel_Number PRIMARY KEY (Hostel_Number)
);

CREATE TABLE OFFENCE_DIM
(
    Offence_Number       NUMBER(5)      NOT NULL,
    Type_Of_Offence      VARCHAR2(1),
    CONSTRAINT PK_STAR_Offence_Number PRIMARY KEY (Offence_Number)
);

CREATE TABLE AUTHOR_DIM
(
    Author_Number        NUMBER(4)      NOT NULL,
    Author_Name          VARCHAR2(40),
    CONSTRAINT PK_STAR_Author_Number PRIMARY KEY (Author_Number)
);

CREATE TABLE SUBJECT_DIM
(
    Subject_Number       NUMBER(5)      NOT NULL,
    Faculty_Number       NUMBER(2)      NOT NULL,
    Subject_Code         VARCHAR2(7),
    CONSTRAINT PK_STAR_Subject_Number PRIMARY KEY (Subject_Number),
    CONSTRAINT FK_STAR_SUBJECT_Faculty_Number FOREIGN KEY (Faculty_Number) REFERENCES
FACULTY_DIM(Faculty_Number)
);

CREATE TABLE BOOK_DIM
```

```
(
  Book_Number          NUMBER(8)          NOT NULL,
  Author_Number        NUMBER(4)          NOT NULL,
  Subject_Number       NUMBER(5)          NOT NULL,
  Book_Edition         VARCHAR2(40),
  CONSTRAINT PK_STAR_Book_Number PRIMARY KEY (Book_Number),
  CONSTRAINT FK_STAR_BD_Author_Number FOREIGN KEY (Author_Number) REFERENCES
AUTHOR_DIM(Author_Number),
  CONSTRAINT FK_STAR_BD_Subject_Number FOREIGN KEY (Subject_Number) REFERENCES
SUBJECT_DIM(Subject_Number)
);
```

CREATE TABLE BOOK_FACT

```
(
  Jaar                 NUMBER(4)          NOT NULL,
  Semester_Number     NUMBER(1)          NOT NULL,
  Maand               NUMBER(2)          NOT NULL,
  Province_Number     NUMBER(2)          NOT NULL,
  Faculty_Number      NUMBER(5)          NOT NULL,
  Hostel_Number       NUMBER(2)          NOT NULL,
  book_type           VARCHAR(1),
  sold_ind            NUMBER(1),
  total_number        NUMBER(5),
  CONSTRAINT PK_STAR_BOOK_FACT PRIMARY KEY (Jaar, Semester_Number, Maand, Province_Number,
Faculty_Number, Hostel_Number),
  CONSTRAINT FK_STAR_BOOK_Province_Number FOREIGN KEY (Province_Number) REFERENCES
PROVINCE_DIM(Province_Number),
  CONSTRAINT FK_STAR_BOOK_Faculty_Number FOREIGN KEY (Faculty_Number) REFERENCES
FACULTY_DIM(Faculty_Number),
  CONSTRAINT FK_STAR_BOOK_Hostel_Number FOREIGN KEY (Hostel_Number) REFERENCES
HOSTEL_DIM(Hostel_Number)
);
```

CREATE TABLE BOOK_PRICE_INFO_FACT

```
(
  Jaar                 NUMBER(4)          NOT NULL,
  Semester_Number     NUMBER(2)          NOT NULL,
  Subject_Number       NUMBER(5)          NOT NULL,
  Average_Price       NUMBER(7,2),
  CONSTRAINT PK_STAR_BPI_FACT PRIMARY KEY (Jaar, Semester_Number, Subject_Number),
  CONSTRAINT FK_STAR_BPI_Subject_Number FOREIGN KEY (Subject_Number) REFERENCES
SUBJECT_DIM(Subject_Number)
);
```

CREATE TABLE BOOK_PRICE_FACT

```
(
  Jaar                 NUMBER(4)          NOT NULL,
  Semester_Number     NUMBER(2)          NOT NULL,
  Book_Number         NUMBER(8)          NOT NULL,
  Average_Price_Successful NUMBER(7,2),
  Average_Price_Unsuccessful NUMBER(7,2),
  CONSTRAINT PK_STAR_BP_FACT PRIMARY KEY (Jaar, Semester_Number, Book_Number),
  CONSTRAINT FK_STAR_BP_Book_Number FOREIGN KEY (Book_Number) REFERENCES BOOK_DIM(Book_Number)
);
```

CREATE TABLE OFFENCE_FACT

```
(
  Jaar                 NUMBER(4)          NOT NULL,
  Semester_Number     NUMBER(1)          NOT NULL,
  Maand               NUMBER(2)          NOT NULL,
  Province_Number     NUMBER(2)          NOT NULL,
  Hostel_Number       NUMBER(2)          NOT NULL,
  Offence_Number      NUMBER(5)          NOT NULL,
  Number_Of_Offences  NUMBER(5),
  CONSTRAINT PK_STAR_O_FACT PRIMARY KEY (Jaar, Semester_Number, Maand, Province_Number, Hostel_Number,
Offence_Number),
  CONSTRAINT FK_STAR_O_Province_Number FOREIGN KEY (Province_Number) REFERENCES
PROVINCE_DIM(Province_Number),
  CONSTRAINT FK_STAR_O_Hostel_Number FOREIGN KEY (Hostel_Number) REFERENCES
HOSTEL_DIM(Hostel_Number),
  CONSTRAINT FK_STAR_O_Offence_Number FOREIGN KEY (Offence_Number) REFERENCES
OFFENCE_DIM(Offence_Number)
);
```

Populations of Facts and Dimensions

Book Fact

```
ALTER TABLE BOOK_FACT
DISABLE CONSTRAINT
    PK_STAR_BOOK_FACT;

ALTER TABLE BOOK_FACT
DISABLE CONSTRAINT
    FK_STAR_BOOK_Province_Number;

ALTER TABLE BOOK_FACT
DISABLE CONSTRAINT
    FK_STAR_BOOK_Faculty_Number;

ALTER TABLE BOOK_FACT
DISABLE CONSTRAINT
    FK_STAR_BOOK_Hostel_Number;

ALTER TABLE SUBJECT_DIM
DISABLE CONSTRAINT
    FK_STAR_SUBJECT_Faculty_Number;

ALTER TABLE FACULTY_DIM
DISABLE CONSTRAINT
    PK_STAR_Faculty_Number;

BEGIN

COMMIT;

SET TRANSACTION READ WRITE NAME 'BookFactUpdate';
SAVEPOINT BookFactSP;

INSERT INTO BOOK_FACT
SELECT DISTINCT
EXTRACT(YEAR FROM b.Date_Placed) "Jaar",
1 AS "Semester",
EXTRACT (MONTH FROM b.Date_Placed) "Maand",
s.Province_Number AS "Province",
Subject_Number AS "Faculty",
s.Hostel_Number AS "Hostel",
SUBSTR(b.Book_Type,1,1),
b.Successful_Sale,
COUNT(b.Book_Number)
FROM BOOK b, STUDENT s, PROVINCE p, HOSTEL h, BOOK_SUBJECT bs
WHERE b.Date_Placed > (SELECT Book_Update_Date FROM BOOK_UPDATE WHERE Book_Update_Number = 1)
AND b.Student_Number = s.Student_Number
AND s.Province_Number = p.Province_Number
AND b.Book_Number = bs.Book_Number
AND s.Hostel_Number = h.Hostel_Number
GROUP BY EXTRACT(YEAR FROM b.Date_Placed), 1, EXTRACT (MONTH FROM b.Date_Placed), p.Province_Number,
bs.Subject_Number, h.Hostel_Number, b.Book_Type, b.Successful_Sale;

UPDATE Book_Fact
SET Semester_Number =2
WHERE maand > 6;

UPDATE Book_Fact
SET Faculty_Number = (SELECT s.Faculty_Number
FROM SUBJECT s
WHERE s.Subject_Number = Book_Fact.Faculty_Number);

INSERT INTO PROVINCE_DIM
SELECT DISTINCT
bf.Province_Number,
p.Province_Name
FROM BOOK_FACT bf, PROVINCE p
WHERE bf.Province_Number = p.Province_Number;

INSERT INTO FACULTY_DIM
SELECT DISTINCT
bf.Faculty_Number,
f.Faculty_Name
FROM BOOK_FACT bf, FACULTY f
```

WHERE bf.Faculty_Number = f.Faculty_Number;

INSERT INTO HOSTEL_DIM
SELECT DISTINCT
bf.Hostel_Number,
h.Hostel_Name
FROM BOOK_FACT bf, HOSTEL h
WHERE bf.Hostel_Number = h.Hostel_Number;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK TO BookFactSP;

END;

/

DELETE FROM PROVINCE_DIM
WHERE ROWID NOT IN
(SELECT MIN(ROWID)
FROM PROVINCE_DIM
GROUP BY Province_Number, Province_Name);

DELETE FROM FACULTY_DIM
WHERE ROWID NOT IN
(SELECT MIN(ROWID)
FROM FACULTY_DIM
GROUP BY Faculty_Number, Faculty_Name);

DELETE FROM HOSTEL_DIM
WHERE ROWID NOT IN
(SELECT MIN(ROWID)
FROM HOSTEL_DIM
GROUP BY Hostel_Number, Hostel_Name);

ALTER TABLE FACULTY_DIM
ENABLE CONSTRAINT
PK_STAR_Faculty_Number;

ALTER TABLE SUBJECT_DIM
ENABLE CONSTRAINT
FK_STAR_SUBJECT_Faculty_Number;

ALTER TABLE BOOK_FACT
ENABLE CONSTRAINT
PK_STAR_BOOK_FACT;

ALTER TABLE BOOK_FACT
ENABLE CONSTRAINT
FK_STAR_BOOK_Province_Number;

ALTER TABLE BOOK_FACT
ENABLE CONSTRAINT
FK_STAR_BOOK_Faculty_Number;

ALTER TABLE BOOK_FACT
ENABLE CONSTRAINT
FK_STAR_BOOK_Hostel_Number;

Book Price Info Fact

```
ALTER TABLE BOOK_FACT
DISABLE CONSTRAINT
    FK_STAR_BOOK_Faculty_Number;

ALTER TABLE BOOK_PRICE_INFO_FACT
DISABLE CONSTRAINT
    PK_STAR_BPI_FACT;

ALTER TABLE BOOK_PRICE_INFO_FACT
DISABLE CONSTRAINT
    FK_STAR_BPI_Subject_Number;

ALTER TABLE BOOK_DIM
DISABLE CONSTRAINT
    FK_STAR_BD_Subject_Number;

ALTER TABLE SUBJECT_DIM
DISABLE CONSTRAINT
    PK_STAR_Subject_Number;

ALTER TABLE SUBJECT_DIM
DISABLE CONSTRAINT
    FK_STAR_SUBJECT_Faculty_Number;

ALTER TABLE FACULTY_DIM
DISABLE CONSTRAINT
    PK_STAR_Faculty_Number;

BEGIN

COMMIT;

SET TRANSACTION READ WRITE NAME 'BookInfoFactUpdate';
SAVEPOINT BookInfoFactSP;

INSERT INTO BOOK_PRICE_INFO_FACT
SELECT DISTINCT
    EXTRACT(YEAR FROM b.Date_Placed) "Jaar",
    EXTRACT (MONTH FROM b.Date_Placed),
    s.Subject_Number AS "Subject",
    AVG(Book_Price) AS "Average Book Price"
FROM BOOK b, SUBJECT s, BOOK_SUBJECT bs
WHERE b.Date_Placed > (SELECT Book_Update_Date FROM BOOK_UPDATE WHERE Book_Update_Number = 1)
AND Successful_Sale = 1
AND b.Book_Number = bs.Book_Number
AND bs.Subject_Number = s.Subject_Number
GROUP BY EXTRACT(YEAR FROM b.Date_Placed), 1, EXTRACT (MONTH FROM b.Date_Placed), s.Subject_Number;

UPDATE BOOK_PRICE_INFO_FACT
SET Semester_Number =2
WHERE Semester_Number > 6;

INSERT INTO SUBJECT_DIM
SELECT DISTINCT
    bf.Subject_Number,
    f.Faculty_Number,
    s.Subject_Code
FROM BOOK_PRICE_INFO_FACT bf, SUBJECT s, FACULTY f
WHERE bf.Subject_Number = s.Subject_Number
AND f.Faculty_Number = s.Faculty_Number;

INSERT INTO FACULTY_DIM
SELECT DISTINCT
    sd.Faculty_Number,
    f.Faculty_Name
FROM SUBJECT_DIM sd, FACULTY f
WHERE sd.Faculty_Number = f.Faculty_Number;

COMMIT;

EXCEPTION

WHEN OTHERS THEN
```



```

ROLLBACK TO BookInfoFactSP;

END;
/
DELETE FROM SUBJECT_DIM
WHERE ROWID NOT IN
(SELECT MIN(ROWID)
FROM SUBJECT_DIM
GROUP BY Subject_Number, Faculty_Number, Subject_Code);

DELETE FROM FACULTY_DIM
WHERE ROWID NOT IN
(SELECT MIN(ROWID)
FROM FACULTY_DIM
GROUP BY Faculty_Number, Faculty_Name);

ALTER TABLE FACULTY_DIM
ENABLE CONSTRAINT
    PK_STAR_Faculty_Number;

ALTER TABLE SUBJECT_DIM
ENABLE CONSTRAINT
    FK_STAR_SUBJECT_Faculty_Number;

ALTER TABLE SUBJECT_DIM
ENABLE CONSTRAINT
    PK_STAR_Subject_Number;

ALTER TABLE BOOK_DIM
ENABLE CONSTRAINT
    FK_STAR_BD_Subject_Number;

ALTER TABLE BOOK_PRICE_INFO_FACT
ENABLE CONSTRAINT
    FK_STAR_BPI_Subject_Number;

ALTER TABLE BOOK_PRICE_INFO_FACT
ENABLE CONSTRAINT
    PK_STAR_BPI_FACT;

ALTER TABLE BOOK_FACT
ENABLE CONSTRAINT
    FK_STAR_BOOK_Faculty_Number;

```

Book Price Fact

```
ALTER TABLE BOOK_FACT
DISABLE CONSTRAINT
    FK_STAR_BOOK_Faculty_Number;

ALTER TABLE BOOK_PRICE_INFO_FACT
DISABLE CONSTRAINT
    FK_STAR_BPI_Subject_Number;

ALTER TABLE BOOK_PRICE_FACT
DISABLE CONSTRAINT
    PK_STAR_BP_FACT;

ALTER TABLE BOOK_PRICE_FACT
DISABLE CONSTRAINT
    FK_STAR_BP_Book_Number;

ALTER TABLE BOOK_DIM
DISABLE CONSTRAINT
    PK_STAR_Book_Number;

ALTER TABLE BOOK_DIM
DISABLE CONSTRAINT
    FK_STAR_BD_Author_Number;

ALTER TABLE BOOK_DIM
DISABLE CONSTRAINT
    FK_STAR_BD_Subject_Number;

ALTER TABLE SUBJECT_DIM
DISABLE CONSTRAINT
    FK_STAR_SUBJECT_Faculty_Number;

ALTER TABLE SUBJECT_DIM
DISABLE CONSTRAINT
    PK_STAR_Subject_Number;

ALTER TABLE FACULTY_DIM
DISABLE CONSTRAINT
    PK_STAR_Faculty_Number;

ALTER TABLE AUTHOR_DIM
DISABLE CONSTRAINT
    PK_STAR_Author_Number;

BEGIN

COMMIT;

SET TRANSACTION READ WRITE NAME 'BookPriceFactUpdate';
SAVEPOINT BookPriceFactSP;

INSERT INTO BOOK_PRICE_FACT
SELECT DISTINCT
EXTRACT(YEAR FROM b.Date_Placed),
1,
b.Book_Number AS "Book Number",
AVG(Book_Price),
0
FROM BOOK b
WHERE b.Date_Placed > (SELECT Book_Update_Date FROM BOOK_UPDATE WHERE Book_Update_Number = 1)
AND EXTRACT (MONTH FROM b.Date_Placed) <= 6
AND Successful_Sale = 1
GROUP BY EXTRACT(YEAR FROM b.Date_Placed), 1, b.Book_Number;

MERGE INTO BOOK_PRICE_FACT f
USING (SELECT * FROM BOOK
WHERE Date_Placed > (SELECT Book_Update_Date FROM BOOK_UPDATE WHERE Book_Update_Number = 1)
AND EXTRACT (MONTH FROM Date_Placed) <= 6
AND Successful_Sale = 0) w
ON ((f.JAAR = EXTRACT(YEAR FROM w.Date_Placed)) AND (f.Semester_Number = 1) AND (f.Book_Number =
w.Book_Number))
WHEN MATCHED THEN UPDATE SET
f.Average_Price_Unsuccessful = (f.Average_Price_Unsuccessful + w.Book_Price)/2
WHEN NOT MATCHED THEN
```

```

INSERT VALUES (EXTRACT (YEAR FROM w.Date_Placed), 1, w.Book_Number, 0, w.Book_Price);

MERGE INTO BOOK_PRICE_FACT f
USING (SELECT * FROM BOOK
WHERE Date_Placed > (SELECT Book_Update_Date FROM BOOK_UPDATE WHERE Book_Update_Number = 1)
AND EXTRACT (MONTH FROM Date_Placed) > 6
AND Successful_Sale = 0) w
ON ((f.JAAR = EXTRACT(YEAR FROM w.Date_Placed)) AND (f.Semester_Number = 2) AND (f.Book_Number =
w.Book_Number))
WHEN MATCHED THEN UPDATE SET
f.Average_Price_Unsuccessful = (f.Average_Price_Unsuccessful + w.Book_Price)/2
WHEN NOT MATCHED THEN
INSERT VALUES (EXTRACT (YEAR FROM w.Date_Placed), 2, w.Book_Number, 0, w.Book_Price);

MERGE INTO BOOK_PRICE_FACT f
USING (SELECT * FROM BOOK
WHERE Date_Placed > (SELECT Book_Update_Date FROM BOOK_UPDATE WHERE Book_Update_Number = 1)
AND EXTRACT (MONTH FROM Date_Placed) > 6
AND Successful_Sale = 1) w
ON ((f.JAAR = EXTRACT(YEAR FROM w.Date_Placed)) AND (f.Semester_Number = 2) AND (f.Book_Number =
w.Book_Number))
WHEN MATCHED THEN UPDATE SET
f.Average_Price_Successful = (f.Average_Price_Successful + w.Book_Price)/2
WHEN NOT MATCHED THEN
INSERT VALUES (EXTRACT (YEAR FROM w.Date_Placed), 2, w.Book_Number, w.Book_Price, 0);

INSERT INTO BOOK_DIM
SELECT DISTINCT
b.Book_Number,
au.Author_Number,
s.Subject_Number,
b.Book_Edition
FROM BOOK b, BOOK_AUTHOR ba, BOOK_SUBJECT bs, AUTHOR au, SUBJECT s
WHERE b.Book_Number = ba.Book_Number
AND ba.Author_Number = au.Author_Number
AND b.Book_Number = bs.Book_Number
AND bs.Subject_Number = s.Subject_Number;

INSERT INTO SUBJECT_DIM
SELECT DISTINCT
bf.Subject_Number,
f.Faculty_Number,
s.Subject_Code
FROM BOOK_DIM bf, SUBJECT s, FACULTY f
WHERE bf.Subject_Number = s.Subject_Number
AND f.Faculty_Number = s.Faculty_Number;

INSERT INTO FACULTY_DIM
SELECT DISTINCT
sd.Faculty_Number,
f.Faculty_Name
FROM SUBJECT_DIM sd, FACULTY f
WHERE sd.Faculty_Number = f.Faculty_Number;

INSERT INTO AUTHOR_DIM
SELECT DISTINCT
bd.Author_Number,
au.Author_Name
FROM BOOK_DIM bd, AUTHOR au
WHERE bd.Author_Number = au.Author_Number;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK TO BookPriceFactSP;

END;
/
DELETE FROM BOOK_DIM
WHERE ROWID NOT IN
(SELECT MIN(ROWID)
FROM BOOK_DIM

```

```
GROUP BY Book_Number, Author_Number, Subject_Number, Book_Edition);
```

```
DELETE FROM SUBJECT_DIM  
WHERE ROWID NOT IN  
(SELECT MIN(ROWID)  
FROM SUBJECT_DIM  
GROUP BY Subject_Number, Faculty_Number, Subject_Code);
```

```
DELETE FROM FACULTY_DIM  
WHERE ROWID NOT IN  
(SELECT MIN(ROWID)  
FROM FACULTY_DIM  
GROUP BY Faculty_Number, Faculty_Name);
```

```
DELETE FROM AUTHOR_DIM  
WHERE ROWID NOT IN  
(SELECT MIN(ROWID)  
FROM AUTHOR_DIM  
GROUP BY Author_Number, Author_Name);
```

```
ALTER TABLE AUTHOR_DIM  
ENABLE CONSTRAINT  
    PK_STAR_Author_Number;
```

```
ALTER TABLE FACULTY_DIM  
ENABLE CONSTRAINT  
    PK_STAR_Faculty_Number;
```

```
ALTER TABLE SUBJECT_DIM  
ENABLE CONSTRAINT  
    PK_STAR_Subject_Number;
```

```
ALTER TABLE SUBJECT_DIM  
ENABLE CONSTRAINT  
    FK_STAR_SUBJECT_Faculty_Number;
```

```
ALTER TABLE BOOK_DIM  
ENABLE CONSTRAINT  
    PK_STAR_Book_Number;
```

```
ALTER TABLE BOOK_DIM  
ENABLE CONSTRAINT  
    FK_STAR_BD_Author_Number;
```

```
ALTER TABLE BOOK_DIM  
ENABLE CONSTRAINT  
    FK_STAR_BD_Subject_Number;
```

```
ALTER TABLE BOOK_PRICE_FACT  
ENABLE CONSTRAINT  
    PK_STAR_BP_FACT;
```

```
ALTER TABLE BOOK_PRICE_FACT  
ENABLE CONSTRAINT  
    FK_STAR_BP_Book_Number;
```

```
ALTER TABLE BOOK_PRICE_INFO_FACT  
ENABLE CONSTRAINT  
    FK_STAR_BPI_Subject_Number;
```

```
ALTER TABLE BOOK_FACT  
ENABLE CONSTRAINT  
    FK_STAR_BOOK_Faculty_Number;
```

Offence Fact

```
ALTER TABLE BOOK_FACT
DISABLE CONSTRAINT
    FK_STAR_BOOK_Province_Number;
```

```
ALTER TABLE BOOK_FACT
DISABLE CONSTRAINT
    FK_STAR_BOOK_Hostel_Number;
```

```
ALTER TABLE OFFENCE_FACT
DISABLE CONSTRAINT
    PK_STAR_O_FACT;
```

```
ALTER TABLE OFFENCE_FACT
DISABLE CONSTRAINT
    FK_STAR_O_Province_Number;
```

```
ALTER TABLE OFFENCE_FACT
DISABLE CONSTRAINT
    FK_STAR_O_Hostel_Number;
```

```
ALTER TABLE OFFENCE_FACT
DISABLE CONSTRAINT
    FK_STAR_O_Offence_Number;
```

```
ALTER TABLE PROVINCE_DIM
DISABLE CONSTRAINT
    PK_STAR_Province_Number;
```

```
ALTER TABLE HOSTEL_DIM
DISABLE CONSTRAINT
    PK_STAR_Hostel_Number;
```

```
ALTER TABLE OFFENCE_DIM
DISABLE CONSTRAINT
    PK_STAR_Offence_Number;
```

```
BEGIN
```

```
COMMIT;
```

```
SET TRANSACTION READ WRITE NAME 'OffenceFactUpdate';
SAVEPOINT OffenceFactSP;
```

```
INSERT INTO OFFENCE_FACT
SELECT DISTINCT
    EXTRACT(YEAR FROM so.Date_Of_Offence) "Jaar",
    1 AS "Semester",
    EXTRACT (MONTH FROM so.Date_Of_Offence) "Maand",
    p.Province_Number AS "Province Number",
    h.Hostel_Number AS "Hostel Number",
    so.Offence_Number AS "Offence Number",
    COUNT(so.Offence_Number) AS "Offences"
FROM STUDENT_OFFENCE so, PROVINCE p, HOSTEL h, STUDENT s
WHERE s.Offence_Number = so.Offence_Number
AND s.Province_Number = p.Province_Number
AND s.Hostel_Number = h.Hostel_Number
GROUP BY EXTRACT(YEAR FROM so.Date_Of_Offence), 1, EXTRACT (MONTH FROM so.Date_Of_Offence),
p.Province_Number, h.Hostel_Number, so.Offence_Number;
```

```
UPDATE OFFENCE_FACT
SET Semester_Number =2
WHERE maand > 6;
```

```
INSERT INTO OFFENCE_DIM
SELECT DISTINCT
    o.Offence_Number,
    so.Type_Of_Offence
FROM OFFENCE_FACT o, STUDENT_OFFENCE so
WHERE o.Offence_Number = so.Offence_Number;
```

```
INSERT INTO PROVINCE_DIM
SELECT DISTINCT
    o.Province_Number,
    p.Province_Name
```

```
FROM OFFENCE_FACT o, PROVINCE p
WHERE o.Province_Number = p.Province_Number;
```

```
INSERT INTO HOSTEL_DIM
SELECT DISTINCT
o.Hostel_Number,
h.Hostel_Name
FROM OFFENCE_FACT o, HOSTEL h
WHERE o.Hostel_Number = h.Hostel_Number;
```

```
COMMIT;
```

```
EXCEPTION
```

```
WHEN OTHERS THEN
```

```
ROLLBACK TO OffenceFactSP;
```

```
END;
```

```
/
```

```
DELETE FROM OFFENCE_DIM
WHERE ROWID NOT IN
(SELECT MIN(ROWID)
FROM OFFENCE_DIM
GROUP BY Offence_Number, Type_Of_Offence);
```

```
DELETE FROM PROVINCE_DIM
WHERE ROWID NOT IN
(SELECT MIN(ROWID)
FROM PROVINCE_DIM
GROUP BY Province_Number, Province_Name);
```

```
DELETE FROM HOSTEL_DIM
WHERE ROWID NOT IN
(SELECT MIN(ROWID)
FROM HOSTEL_DIM
GROUP BY Hostel_Number, Hostel_Name);
```

```
ALTER TABLE OFFENCE_DIM
ENABLE CONSTRAINT
PK_STAR_Offence_Number;
```

```
ALTER TABLE HOSTEL_DIM
ENABLE CONSTRAINT
PK_STAR_Hostel_Number;
```

```
ALTER TABLE PROVINCE_DIM
ENABLE CONSTRAINT
PK_STAR_Province_Number;
```

```
ALTER TABLE OFFENCE_FACT
ENABLE CONSTRAINT
PK_STAR_O_FACT;
```

```
ALTER TABLE OFFENCE_FACT
ENABLE CONSTRAINT
FK_STAR_O_Province_Number;
```

```
ALTER TABLE OFFENCE_FACT
ENABLE CONSTRAINT
FK_STAR_O_Hostel_Number;
```

```
ALTER TABLE OFFENCE_FACT
ENABLE CONSTRAINT
FK_STAR_O_Offence_Number;
```

```
ALTER TABLE BOOK_FACT
ENABLE CONSTRAINT
FK_STAR_BOOK_Hostel_Number;
```

```
ALTER TABLE BOOK_FACT
ENABLE CONSTRAINT
FK_STAR_BOOK_Province_Number;
```

Creation Of Indexes

```
DROP INDEX OFFENCE_TYPE_IDX;  
DROP INDEX BOOK_TYPE_IDX;
```

```
CREATE INDEX OFFENCE_TYPE_IDX  
ON STUDENT_OFFENCE (Type_Of_Offence);
```

```
CREATE INDEX BOOK_TYPE_IDX  
ON BOOK (Book_Type);
```

User Views

/*

```
DROP VIEW BOOK_FACT_DETAIL;  
DROP VIEW BOOK_PRICE_INFO_FACT_DETAIL;  
DROP VIEW BOOK_PRICE_FACT_DETAIL;  
DROP VIEW OFFENCE_FACT_DETAIL;
```

*/

CREATE OR REPLACE VIEW BOOK_FACT_DETAIL

```
AS SELECT Jaar "Year", Semester_Number "Semester", Maand "Month", Province_Name "Province", Faculty_Name  
"Faculty", Hostel_Name "Hostel", Book_Type "Type", Sold_Ind "Successful", Total_Number "Total"  
FROM BOOK_FACT  
NATURAL JOIN PROVINCE_DIM  
NATURAL JOIN FACULTY_DIM  
NATURAL JOIN HOSTEL_DIM;
```

CREATE OR REPLACE VIEW BOOK_PRICE_INFO_FACT_DETAIL

```
AS SELECT Jaar "Year", Semester_Number "Semester", Subject_Code "Subject", Average_Price "Average Price"  
FROM BOOK_PRICE_INFO_FACT  
NATURAL JOIN SUBJECT_DIM;
```

CREATE OR REPLACE VIEW BOOK_PRICE_FACT_DETAIL

```
AS SELECT Jaar "Year", Semester_Number "Semester", Subject_Code "Subject", Author_Name "Author", Book_Edition  
"Edition", Average_Price_Successful "Average Price Sold", Average_Price_Unsuccessful "Average Price Unsold"  
FROM BOOK_PRICE_FACT  
NATURAL JOIN BOOK_DIM  
NATURAL JOIN AUTHOR_DIM  
NATURAL JOIN SUBJECT_DIM;
```

CREATE OR REPLACE VIEW OFFENCE_FACT_DETAIL

```
AS SELECT Jaar "Year", Semester_Number "Semester", Maand "Month", Province_Name "Province", Hostel_Name "Hostel",  
Type_Of_Offence "Offence", Number_Of_Offences "Total Offences"  
FROM OFFENCE_FACT  
NATURAL JOIN HOSTEL_DIM  
NATURAL JOIN OFFENCE_DIM  
NATURAL JOIN PROVINCE_DIM;
```


Update Of Tables

```
DROP TABLE BOOK_UPDATE;  
DROP TABLE OFFENCE_UPDATE;
```

```
PURGE TABLE BOOK_UPDATE;  
PURGE TABLE OFFENCE_TABLE;
```

```
CREATE TABLE BOOK_UPDATE  
(  
    Book_Update_Number      NUMBER(1)      NOT NULL,  
    Book_Update_Date        DATE,  
    CONSTRAINT PK_BOOK_UPDATE PRIMARY KEY (Book_Update_Number)  
);
```

```
CREATE TABLE OFFENCE_UPDATE  
(  
    Offence_Update_Number   NUMBER(1)      NOT NULL,  
    Offence_Update_Date     DATE,  
    CONSTRAINT PK_OFFENCE_UPDATE PRIMARY KEY (Offence_Update_Number)  
);
```

```
INSERT ALL  
INTO BOOK_UPDATE(Book_Update_Number, Book_Update_Date) VALUES(1, TO_DATE('01-Jan-18','DD-MON-YY'))  
INTO OFFENCE_UPDATE(Offence_Update_Number, Offence_Update_Date) VALUES(1, TO_DATE('01-Jan-18','DD-MON-YY'))  
SELECT 1 FROM DUAL;
```

```
UPDATE BOOK_UPDATE  
SET Book_Update_Number = 1, Book_Update_Date = SYSDATE;
```

```
UPDATE OFFENCE_UPDATE  
SET Offence_Update_Number = 1, Offence_Update_Date = SYSDATE;
```