TIS1101 Database Fundamentals / TDB2111 Database Systems

Group Project

Title: Concert Ticketing System

Prepared by:

Leader : 1141128015 Jeffrey Tan Kah Jun jeffreytkj96@gmail.com

Member: 1141128589 Yap Yung Seng ysyap1314@gmail.com

Member: 1141128324 Ng Wee Fon ngwfepm@gmail.com

Member: 1141128390 Kevin Toh terrible988765@gmail.com

Table of Contents

1.0 Corrected Business Rule

- A sales location can be worked by many cashiers, and each cashier can only work at one sales location.

- A cashier may serves several customers, and a customer can only be served by one cashier only.

- Each customer can buy more than 1 tickets, each ticket can be bought by only 1 customer.

- A concert showing uses many tickets, but one ticket can only be used in one concert showing.

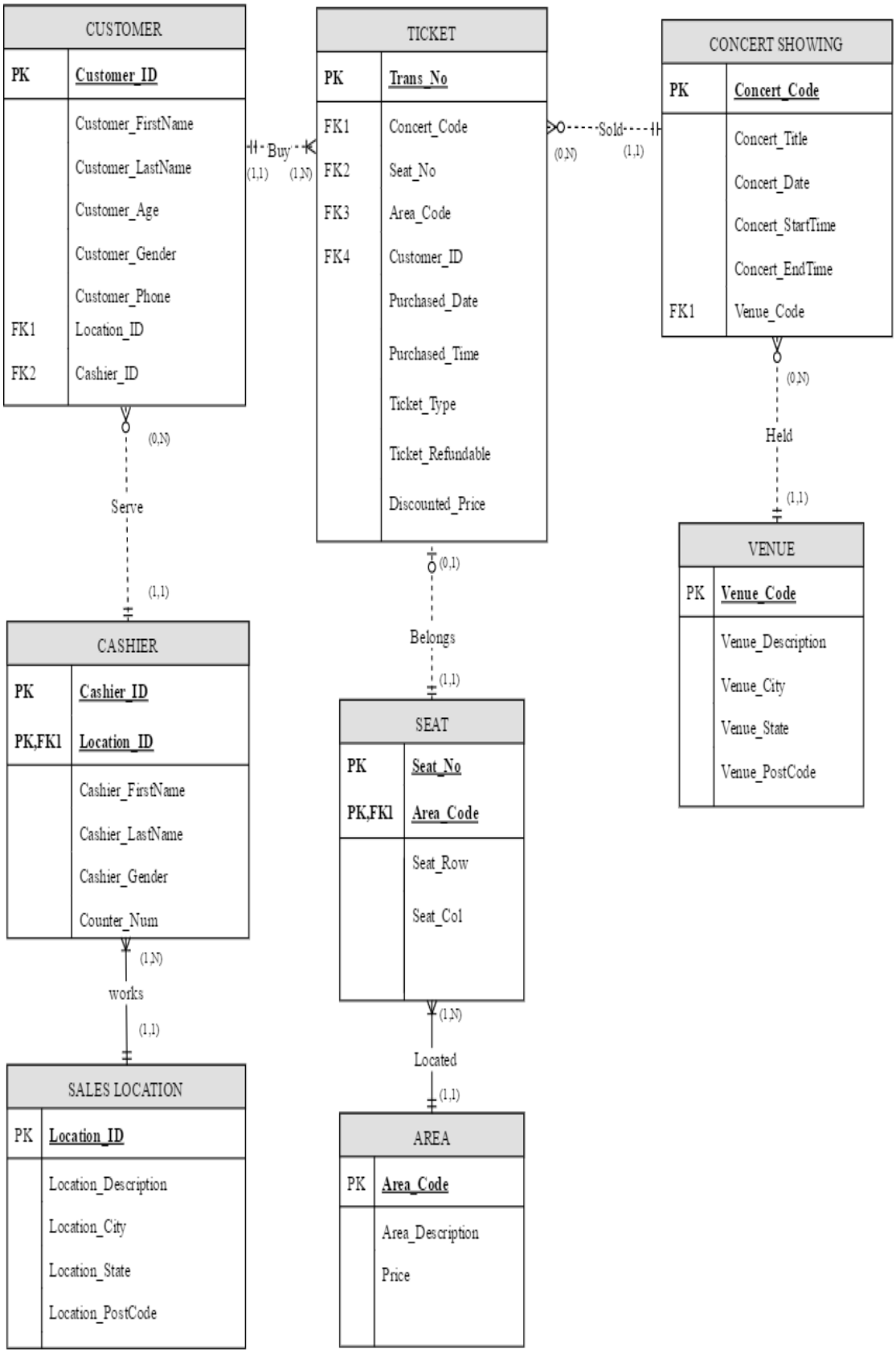
- One venue may held several concert at different dates, and each concert can only be held at one venue.

- Each ticket may belong to only 1 seat, but each seat may belong to only 1 ticket.

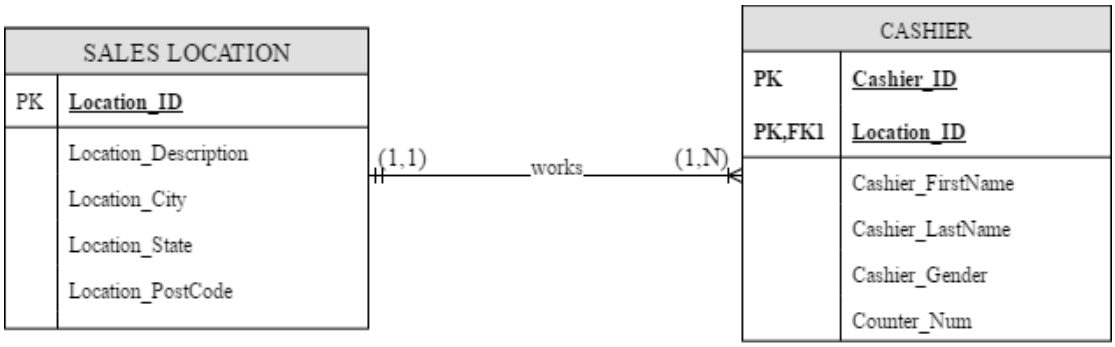
- Each seat is located in one area but each area contains many seats.

- If customer wish to cancel the ticket and refund it must be done before 7 days of the concert showing begins.

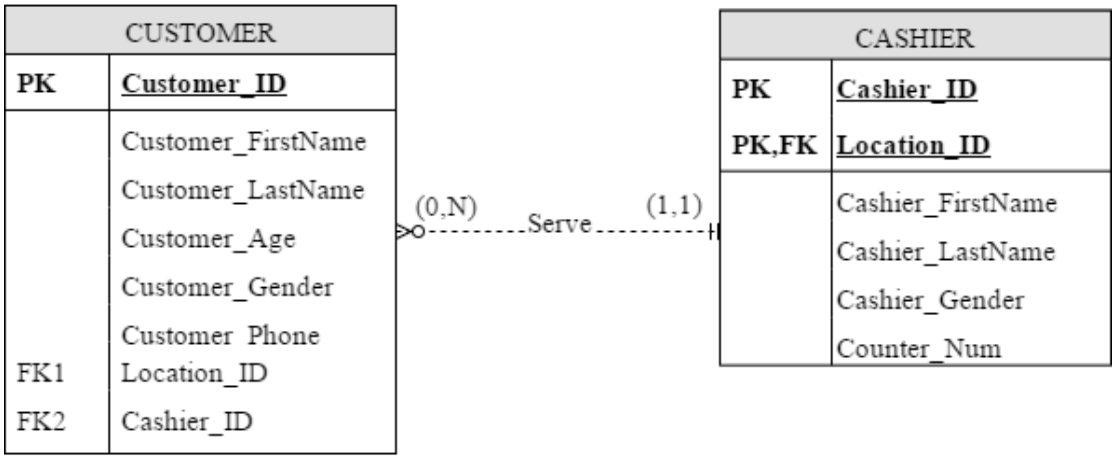
2.0 Corrected Entity-Relationship-Diagram



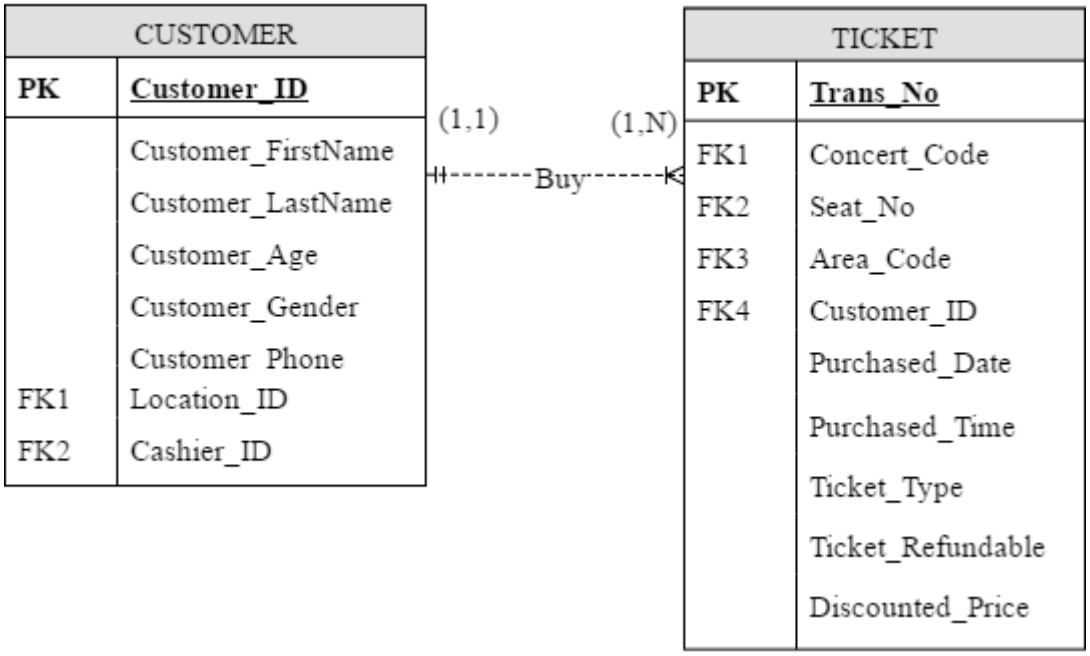
3.0 Relationships



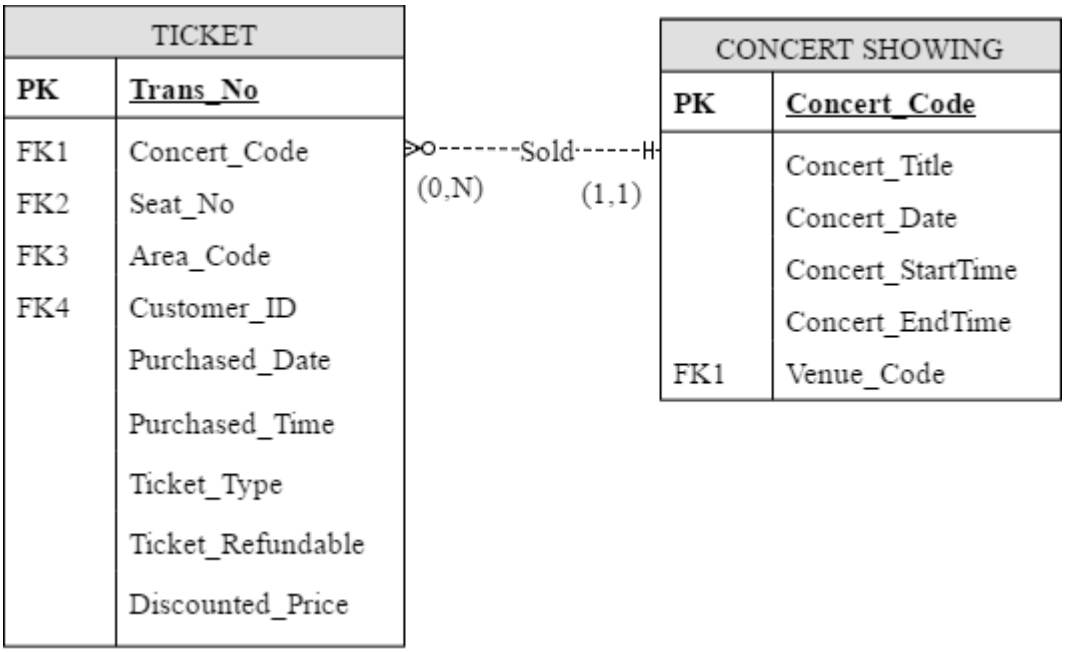
A sales location can be worked by many cashiers, and each cashier can only work at one sales location. Therefore, it is a 1:M relationship. It is also a Mandatory Participation.



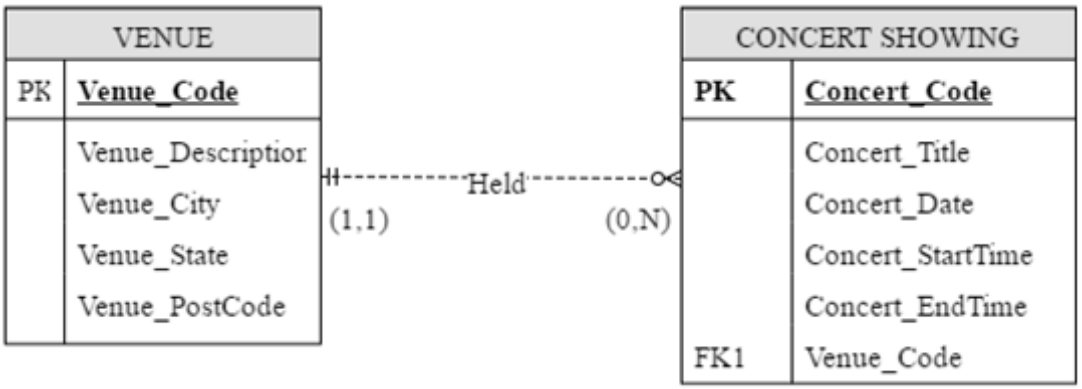
One cashier may serve many customer, and each customer will only be served by one cashier. Therefore, it is a 1:M relationship. It is also an Optional Participation.



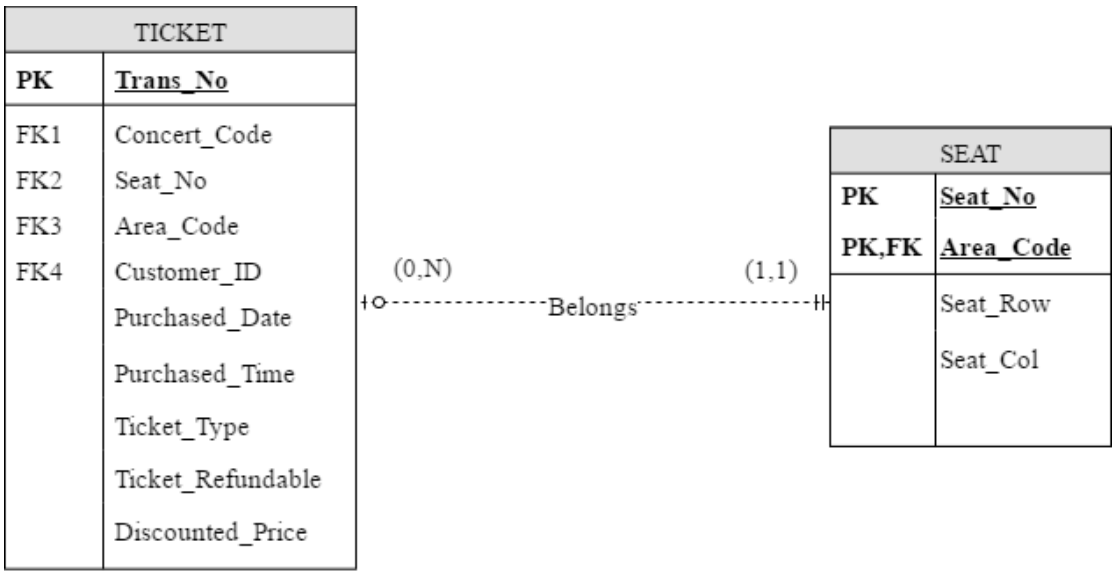
One customer can buy many tickets, but one ticket can only be bought by one customer. Therefore, it is a 1:M relationship. It is also a Mandatory Participation.



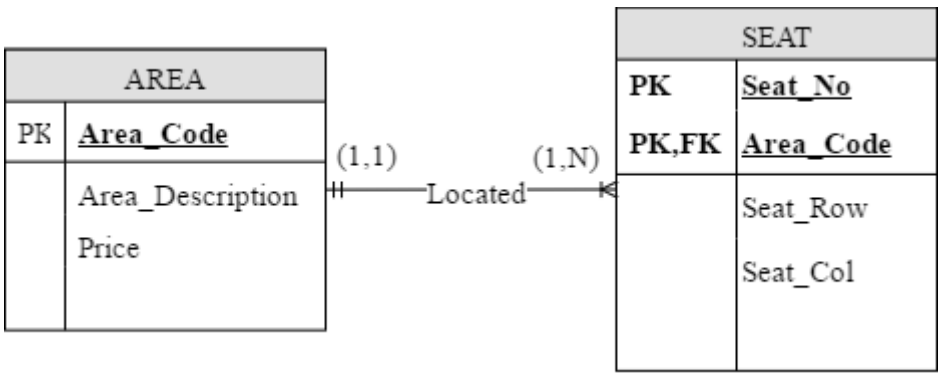
One venue may held many concerts, and every concert can only be held at one venue. Therefore, it is a 1:M relationship. It is also an Optional Participation.



One venue may held many concerts, and every concert can only be held at one venue at different day. Therefore, it is a 1:M relationship. It is also an Optional Participation.



One seat may belong to one ticket, but each ticket belongs to one seat only. Therefore, it is a 1:1 relationship. It is also an Optional Participation.



One area contains many seats, and one seat only located in one area. Therefore, it is a 1:M relationship. It is also an Mandatory Participation.

4.0 Data Definition Command (DDL)

```
CREATE TABLE SALES_LOCATION(
        Location_ID integer NOT NULL GENERATED ALWAYS AS IDENTITY,
        Location_Description varchar(20) NOT NULL,
        Location_City varchar(20),
        Location_State  varchar(20),
        Location_PostCode varchar(5) ,
        PRIMARY KEY(Location_ID),
        CONSTRAINT Location_Unq UNIQUE(Location_Description)
)@

CREATE TABLE CASHIER(
        Cashier_ID varchar(5) NOT NULL,
        Location_ID integer NOT NULL,
        Cashier_FirstName varchar(20) NOT NULL,
        Cashier_LastName varchar(20) NOT NULL,
        Cashier_Gender varchar(1) CHECK(Cashier_Gender IN('F','M')) NOT NULL,
        Counter_Num varchar(1) NOT NULL,
        PRIMARY KEY(Cashier_ID,Location_ID),
        FOREIGN KEY(Location_ID) REFERENCES SALES_LOCATION ON DELETE CASCADE,
        CONSTRAINT  Counter_Unq UNIQUE(Location_ID, Counter_Num)
)@

CREATE TABLE CUSTOMER(
        Customer_ID varchar(6) NOT NULL,
        Customer_FirstName varchar(20) NOT NULL,
        Customer_LastName varchar(20) NOT NULL,
        Customer_Age integer,
        Customer_Gender varchar(1) CHECK(Customer_Gender IN('F','M')) ,
        Customer_Phone varchar(11),
        Cashier_ID varchar(5) ,
        Location_ID integer,
        PRIMARY KEY(Customer_ID),
        FOREIGN KEY(Cashier_ID,Location_ID) REFERENCES CASHIER ON DELETE SET NULL
)@

CREATE TABLE AREA(
        Area_Code varchar(3) NOT NULL,
        Area_Description varchar(10) NOT NULL,
        Price decimal(5,2) NOT NULL,
        PRIMARY KEY(Area_Code),
        CONSTRAINT Area_Unq UNIQUE(Area_Description)
)@
```

```
CREATE TABLE SEAT(
        Seat_No varchar(6) NOT NULL,
        Area_Code varchar(3) NOT NULL,
        Seat_Row varchar(2) NOT NULL,
        Seat_Col varchar(2) NOT NULL,
        PRIMARY KEY(Seat_No,Area_Code),
        FOREIGN KEY(Area_Code) REFERENCES AREA ON DELETE RESTRICT,
        CONSTRAINT Seat_Unq UNIQUE(Area_Code,Seat_Row,Seat_Col)
)@

CREATE TABLE VENUE(
        Venue_Code varchar(3) NOT NULL,
        Venue_Description varchar(20) NOT NULL,
        Venue_City varchar(20),
        Venue_State varchar(20),
        Venue_PostCode varchar(5),
        PRIMARY KEY(Venue_Code),
        CONSTRAINT Venue_Unq Unique(Venue_Description)
)@

CREATE TABLE CONCERT_SHOWING(
        Concert_Code varchar(7) NOT NULL,
        Concert_Title varchar(20) NOT NULL,
        Concert_Date date NOT NULL,
        Concert_StartTime time NOT NULL,
        Concert_EndTime time NOT NULL,
        Venue_Code varchar(3) NOT NULL,
        PRIMARY KEY(Concert_Code),
        FOREIGN KEY(Venue_Code) REFERENCES VENUE ON DELETE RESTRICT,
        CONSTRAINT Concert_Unq UNIQUE(Concert_Date, Venue_Code)
)@

CREATE TABLE TICKET(
        Trans_No varchar(8) NOT NULL,
        Concert_Code varchar(7) NOT NULL,
        Seat_No varchar(6) NOT NULL,
        Area_Code varchar(3) NOT NULL,
        Customer_ID varchar(6) NOT NULL,
        Purchased_Date date NOT NULL,
        Purchased_Time time NOT NULL,
        Ticket_Type varchar(20) CHECK(Ticket_Type IN('Adult','Children')) NOT NULL,
        Ticket_Refundable varchar(20) DEFAULT'Available',
        Discounted_Price decimal(5,2),
        PRIMARY KEY(Trans_No),
        FOREIGN KEY(Concert_Code) REFERENCES CONCERT_SHOWING ON DELETE CASCADE,
        FOREIGN KEY(Seat_No,Area_Code) REFERENCES SEAT ON DELETE RESTRICT,
        FOREIGN KEY(Customer_ID) REFERENCES CUSTOMER ON DELETE RESTRICT,
        CONSTRAINT Seaq_Unq UNIQUE(Concert_Code, Seat_No, Area_Code)
)@
```

5.0 Data Insertion

Insertion on Sales_Location table

```
INSERT INTO SALES_LOCATION VALUES('IOI Mall','Puchong','Selangor','47100')@
INSERT INTO SALES_LOCATION VALUES('TESCO','Puchong','SELANGOR','47100')@
INSERT INTO SALES_LOCATION VALUES('Setia Walk','Puchong','Selangor','47100')@
INSERT INTO SALES_LOCATION VALUES('Paradigm','Petaling Jaya','Selangor','60000')@
INSERT INTO SALES_LOCATION VALUES('Mid Valley','Kepong','Kuala Lumpur','58000')@
```

```
db2 => select * from sales_location@

LOCATION_ID LOCATION_DESCRIPTION LOCATION_CITY        LOCATION_STATE       LOCATION_POSTCODE
----------- -------------------- -------------------- -------------------- ----------------
          1 IOI Mall             Puchong              Selangor             47100
          2 TESCO                Puchong              Selangor             47100
          3 Setia Walk           Puchong              Selangor             47100
          4 Paradigm             Petaling Jaya        Selangor             60000
          5 Mid Valley           Kepong               Kuala Lumpur         58000
```

Insertion on Cashier table

```
INSERT INTO CASHIER VALUES('C0001',1,'Justin','Ng','M','1')@
INSERT INTO CASHIER VALUES('C0002',1,'Michele','Hee','F','2')@
INSERT INTO CASHIER VALUES('C0003',2,'Xyres','Ng','M','1')@
INSERT INTO CASHIER VALUES('C0004',2,'Kevin','Liew','M','2')@
INSERT INTO CASHIER VALUES('C0005',3,'Selina','Ho','F','1')@
INSERT INTO CASHIER VALUES('C0006',4,'Donnie','Ho','F','1')@
INSERT INTO CASHIER VALUES('C0007',5,'Donnie','Ho','F','1')@
INSERT INTO CASHIER VALUES('C0008',5,'Sean','Chow','F','2')@
INSERT INTO CASHIER VALUES('C0009',5,'Bob','Yee','F','3')@
```

```
db2 => select * from Cashier@

CASHIER_ID LOCATION_ID CASHIER_FIRSTNAME    CASHIER_LASTNAME    CASHIER_GENDER COUNTER_NUM
---------- ----------- -------------------- -------------------- -------------- -----------
C0001                1 Justin               Ng                   M                        1
C0002                1 Michele              Hee                  F                        2
C0003                2 Xyres                Ng                   M                        1
C0004                2 Kevin                Liew                 M                        2
C0005                3 Selina               Ho                   F                        1
C0006                4 Donnie               Ho                   F                        1
C0007                5 Donnie               Ho                   F                        1
C0008                5 Sean                 Chow                 F                        2
C0009                5 Bob                  Yee                  F                        3
```

Insertion on Customer table

INSERT INTO CUSTOMER VALUES('000000','John','Liew',20,'M','01225698745','C0001',1)@
INSERT INTO CUSTOMER VALUES('000001','Jason','Loh',13,'M','0145986369','C0002',1)@
INSERT INTO CUSTOMER VALUES('000002','Nicole','Teh',20,'F','032596554','C0002',1)@
INSERT INTO CUSTOMER VALUES('000003','Jeff','Liew',20,'M','01263549738','C0003',2)@
INSERT INTO CUSTOMER VALUES('000004','Nick','Loh',13,'M','01827403648','C0004',2)@
INSERT INTO CUSTOMER VALUES('000005','Andrew','Teh',20,'F','038052395','C0008',5)@
INSERT INTO CUSTOMER VALUES('000006','Yi gor','Teh',20,'M','038052328','C0008',5)@

```
db2 => select * from customer@

CUSTOMER_ID CUSTOMER_FIRSTNAME   CUSTOMER_LASTNAME    CUSTOMER_AGE CUSTOMER_GENDER CUSTOMER_PHONE CASHIER_ID LOCATION_ID
----------- ------------------ ------------------ ------------ --------------- -------------- ---------- -----------
000000      John               Liew                         20 M               01225698745    C0001               1
000001      Jason              Loh                          13 M               0145986369     C0002               1
000002      Nicole             Teh                          20 F               032596554      C0002               1
000003      Jeff               Liew                         20 M               01263549738    C0003               2
000004      Nick               Loh                          13 M               01827403648    C0004               2
000005      Andrew             Teh                          20 F               038052395      C0008               5
000006      Yi gor             Teh                          20 M               038052328      C0008               5
```

Insertion on Area table

INSERT INTO AREA VALUES('V1','VIP','600')@
INSERT INTO AREA VALUES('V2','VVIP','700')@
INSERT INTO AREA VALUES('R1','ROCK A','500')@
INSERT INTO AREA VALUES('R2','ROCK B','450')@
INSERT INTO AREA VALUES('N','Normal','200')@

```
db2 => select * from area@

AREA_CODE AREA_DESCRIPTION PRICE
--------- ---------------- --------
V1        VIP               600.00
V2        VVIP              700.00
R1        ROCK A            500.00
R2        ROCK B            450.00
N         Normal            200.00
```

Insertion on Seat table

```
INSERT INTO SEAT VALUES('1','V1','1','1')@
INSERT INTO SEAT VALUES('2','V1','1','2')@
INSERT INTO SEAT VALUES('3','V1','1','3')@
INSERT INTO SEAT VALUES('4','V1','1','4')@
INSERT INTO SEAT VALUES('5','V1','1','5')@
INSERT INTO SEAT VALUES('1','V2','1','1')@
INSERT INTO SEAT VALUES('2','V2','1','2')@
INSERT INTO SEAT VALUES('3','V2','1','3')@
INSERT INTO SEAT VALUES('4','V2','1','4')@
INSERT INTO SEAT VALUES('5','V2','1','5')@
INSERT INTO SEAT VALUES('1','R1','1','1')@
INSERT INTO SEAT VALUES('2','R1','1','2')@
INSERT INTO SEAT VALUES('3','R1','1','3')@
INSERT INTO SEAT VALUES('1','R2','1','1')@
INSERT INTO SEAT VALUES('2','R2','1','2')@
INSERT INTO SEAT VALUES('3','R2','1','3')@
INSERT INTO SEAT VALUES('4','R2','1','4')@
INSERT INTO SEAT VALUES('1','N','1','1')@
INSERT INTO SEAT VALUES('2','N','1','2')@
INSERT INTO SEAT VALUES('3','N','1','3')@
```

```
db2 => select * from seat@

SEAT_NO AREA_CODE SEAT_ROW SEAT_COL
------- --------- -------- --------
1       V1        1        1
2       V1        1        2
3       V1        1        3
4       V1        1        4
5       V1        1        5
1       V2        1        1
2       V2        1        2
3       V2        1        3
4       V2        1        4
5       V2        1        5
1       R1        1        1
2       R1        1        2
3       R1        1        3
1       R2        1        1
2       R2        1        2
3       R2        1        3
4       R2        1        4
1       N         1        1
2       N         1        2
3       N         1        3
```

Insertion on Venue table

INSERT INTO VENUE VALUES('BJ','KL Bukit Jalil','Sepang','Kuala Lumpur','60000')@
INSERT INTO VENUE VALUES('SM','Stadium Merdeka','Kepong','Kuala Lumpur','50150')@
INSERT INTO VENUE VALUES('G','Genting','Puchong','Selangor','50150')@

```
db2 => select * from venue@

VENUE_CODE VENUE_DESCRIPTION    VENUE_CITY          VENUE_STATE         VENUE_POSTCODE
---------- -------------------- ------------------- ------------------- --------------
BJ         KL Bukit Jalil       Sepang              Kuala Lumpur        60000
SM         Stadium Merdeka      Kepong              Kuala Lumpur        50150
G          Genting              Puchong             Selangor            50150
```

Insertion on Concert_Showing table

INSERT INTO CONCERT_SHOWING VALUES('J','Jay Tour',
'02/27/2017','18:00:00','22:00:00','BJ')@

INSERT INTO CONCERT_SHOWING VALUES('B','Beyond Tour',
'02/26/2017','18:00:00','22:00:00','BJ')@

INSERT INTO CONCERT_SHOWING VALUES('G','GEM Tour',
'02/27/2017','18:00:00','22:00:00','SM')@

INSERT INTO CONCERT_SHOWING VALUES('A','Alice Tour',
'02/22/2017','18:00:00','22:00:00','SM')@

INSERT INTO CONCERT_SHOWING VALUES('JT','Jeff Tan Tour',
'03/15/2017','18:00:00','22:00:00','G')@

INSERT INTO CONCERT_SHOWING VALUES('W','Lee horm Tour',
'03/16/2017','18:00:00','22:00:00','SM')@

INSERT INTO CONCERT_SHOWING VALUES('WW','Wee World Tour',
'02/07/2017','18:00:00','22:00:00','SM')@

INSERT INTO CONCERT_SHOWING VALUES('JW','Jack World Tour',
'02/08/2017','18:00:00','22:00:00','BJ')@

```
db2 => select * from concert_showing@

CONCERT_CODE CONCERT_TITLE        CONCERT_DATE CONCERT_STARTTIME CONCERT_ENDTIME VENUE_CODE
------------ -------------------- ------------ ---------------- -------------- ----------
J            Jay Chou Tour        27-02-2017   18:00:00         22:00:00       BJ
B            Beyond Tour          26-02-2017   18:00:00         22:00:00       BJ
G            GEM Tour             27-02-2017   18:00:00         22:00:00       SM
A            Alice Tour           22-02-2017   18:00:00         22:00:00       SM
JT           Jeff Tan Tour        15-03-2017   18:00:00         22:00:00       G
W            Lee horm Tour        16-03-2017   18:00:00         22:00:00       SM
WW           Wee World Tour       07-02-2017   18:00:00         22:00:00       SM
JW           Jack World Tour      08-02-2017   18:00:00         22:00:00       BJ
```

Insertion on Ticket table (Using Store Procedure)

call InsertTicket('T1','J','1','V1','000001',CURRENT TIMESTAMP,CURRENT TIMESTAMP,'Adult')@

call InsertTicket('T2','J','2','V1','000001',CURRENT TIMESTAMP,CURRENT TIMESTAMP,'Children')@

call InsertTicket('T3','J','1','V2','000002',CURRENT TIMESTAMP,CURRENT TIMESTAMP,'Adult')@

call InsertTicket('T4','G','1','R1','000002',CURRENT TIMESTAMP,CURRENT TIMESTAMP,'Adult')@

call InsertTicket('T5','G','2','R1','000002',CURRENT TIMESTAMP,CURRENT TIMESTAMP,'Children')@

call InsertTicket('T6','W','1','R1','000003',CURRENT TIMESTAMP,CURRENT TIMESTAMP,'Adult')@

call InsertTicket('T7','B','3','R1','000004',CURRENT TIMESTAMP,CURRENT TIMESTAMP,'Children')@

call InsertTicket('T8','B','2','N','000005',CURRENT TIMESTAMP,CURRENT TIMESTAMP,'Children')@

call InsertTicket('T9','WW','1','V1','000006',CURRENT TIMESTAMP,CURRENT TIMESTAMP,'Children')@

call InsertTicket('T10','JW','1','V2','000003',CURRENT TIMESTAMP,CURRENT TIMESTAMP,'Adult')@

```
db2 => select * from ticket@

TRANS_NO CONCERT_CODE SEAT_NO AREA_CODE CUSTOMER_ID PURCHASED_DATE PURCHASED_TIME TICKET_TYPE         TICKET_REFUNDABLE    DISCOUNTED_PRICE
-------- ------------ ------- --------- ----------- ------------- ------------- ------------------- -------------------- ----------------
T1       J            1       V1        000001      07-02-2017    00:36:13      Adult               Available                       600.00
T2       J            2       V1        000001      07-02-2017    00:36:13      Children            Available                       480.00
T3       J            1       V2        000002      07-02-2017    00:36:13      Adult               Available                       700.00
T4       G            1       R1        000002      07-02-2017    00:36:13      Adult               Available                       500.00
T5       G            2       R1        000002      07-02-2017    00:36:13      Children            Available                       400.00
T6       W            1       R1        000003      07-02-2017    00:36:13      Adult               Available                       500.00
T7       B            3       R1        000004      07-02-2017    00:36:13      Children            Available                       400.00
T8       B            2       N         000005      07-02-2017    00:36:13      Children            Available                       160.00
T9       WW           1       V1        000006      07-02-2017    00:36:13      Children            Available                       480.00
T10      JW           1       V2        000003      07-02-2017    00:36:13      Adult               Available                       700.00
```

6.0 Data Manipulation Language (DML)

6.1 Aggregate Function

Total income from a selected concert

```
CREATE FUNCTION getIncomeConcert(ConcertCode varchar(7))

RETURNS TABLE(Concert_Code varchar(7), Concert_Title varchar(20), discounted_Price
decimal(20,2) )

RETURN

        SELECT TICKET.Concert_Code, CONCERT_SHOWING.Concert_Title

        , SUM(Discounted_Price) AS Total_Income

        FROM TICKET, CONCERT_SHOWING

        WHERE TICKET.Concert_Code=CONCERT_SHOWING.Concert_Code

        AND TICKET.Concert_Code=ConcertCode

        GROUP BY TICKET.Concert_Code, CONCERT_SHOWING.Concert_Title@
```

```
db2 => select * from table(getIncomeConcert('J'))@

CONCERT_CODE CONCERT_TITLE        DISCOUNTED_PRICE
------------ -------------------- ----------------------
J            Jay Chou Tour                      1780.00

  1 record(s) selected.

db2 => select * from table(getIncomeConcert('G'))@

CONCERT_CODE CONCERT_TITLE        DISCOUNTED_PRICE
------------ -------------------- ----------------------
G            GEM Tour                            900.00
```

Area which has the highest price

```
SELECT Area_Code, Area_Description, Price
FROM AREA
WHERE Price=(SELECT MAX(Price) FROM AREA)@
```

```
db2 => SELECT Area_Code, Area_Description, Price
db2 (cont.) => FROM AREA
db2 (cont.) => WHERE Price=(SELECT MAX(Price)
db2 (cont.) =>       FROM AREA)@

AREA_CODE AREA_DESCRIPTION PRICE
--------- ---------------- -------
V2        VVIP              700.00
```

Area which has the lowest price

SELECT Area_Code, Area_Description, Price
FROM AREA
WHERE Price=(SELECT MIN(Price) FROM AREA)@

```
db2 => SELECT Area_Code, Area_Description, Price
db2 (cont.) => FROM AREA
db2 (cont.) => WHERE Price=(SELECT MAX(Price)
db2 (cont.) =>       FROM AREA)@

AREA_CODE AREA_DESCRIPTION PRICE
--------- ---------------- -------
V2        VVIP              700.00
```

6.2 Query with a group by and having clauses

Customers who purchased more than 2 tickets

SELECT Customer_ID, COUNT(Trans_No) AS Number_Of_Ticket

FROM TICKET

GROUP BY Customer_ID

HAVING COUNT(Trans_No) > 2@

```
db2 => SELECT Customer_ID, COUNT(Trans_No) AS Number_Of_Ticket
db2 (cont.) => FROM TICKET
db2 (cont.) => GROUP BY Customer_ID
db2 (cont.) => HAVING COUNT(Trans_No) > 2@

CUSTOMER_ID NUMBER_OF_TICKET
----------- ----------------
000002                    3
```

The number of ticket that is bought by each customers

SELECT Customer_ID, COUNT(Trans_No) AS Number_Of_Ticket

FROM TICKET

GROUP BY Customer_ID@

```
db2 => SELECT Customer_ID, COUNT(Trans_No) AS Number_Of_Ticket
db2 (cont.) => FROM TICKET
db2 (cont.) => GROUP BY Customer_ID@

CUSTOMER_ID NUMBER_OF_TICKET
----------- ----------------
000001                    2
000002                    3
000003                    2
000004                    1
000005                    1
000006                    1
```

The total price of tickets that customer have bought

SELECT Customer_ID, SUM(Discounted_Price) AS Total_Price

FROM TICKET

GROUP BY Customer_ID@

```
db2 => SELECT Customer_ID, SUM(Discounted_Price) AS Total_Price
db2 (cont.) => FROM TICKET
db2 (cont.) => GROUP BY Customer_ID@

CUSTOMER_ID TOTAL_PRICE
----------- ---------------------------------
000001                              1080.00
000002                              1600.00
000003                              1200.00
000004                               400.00
000005                               160.00
000006                               480.00
```

The total price of tickets that customer have bought is less than 300 dolar

SELECT Customer_ID, SUM(Discounted_Price) AS Total_Price

FROM TICKET

GROUP BY Customer_ID

HAVING SUM(Discounted_Price) < 300@

```
db2 => SELECT Customer_ID, SUM(Discounted_Price) AS Total_Price
db2 (cont.) => FROM TICKET
db2 (cont.) => GROUP BY Customer_ID
db2 (cont.) => HAVING SUM(Discounted_Price) < 300@

CUSTOMER_ID TOTAL_PRICE
----------- ---------------------------------
000005                                 160.00
```

The number of seat in each area

SELECT SEAT.Area_Code, Area_Description, COUNT(Seat_No) AS Number_Of_Seat

FROM SEAT,AREA

WHERE AREA.Area_Code=SEAT.Area_Code

GROUP BY SEAT.Area_Code, AREA.Area_Description@

```
db2 => SELECT SEAT.Area_Code, Area_Description, COUNT(Seat_No) AS Number_Of_Seat
db2 (cont.) => FROM SEAT,AREA
db2 (cont.) => WHERE AREA.Area_Code=SEAT.Area_Code
db2 (cont.) => GROUP BY SEAT.Area_Code, AREA.Area_Description@

AREA_CODE AREA_DESCRIPTION NUMBER_OF_SEAT
--------- --------------- -------------
U1        VIP                         5
U2        VVIP                        5
R1        ROCK A                      3
R2        ROCK B                      4
N         Normal                      3
```

Total income for each concert showing

SELECTTICKET.Concert_Code, CONCERT_SHOWING.Concert_Title, SUM(Discounted_Price) AS Total_Income

FROM TICKET,CONCERT_SHOWING

WHERE TICKET.Concert_Code=CONCERT_SHOWING.Concert_Code

GROUP BY TICKET.Concert_Code,CONCERT_SHOWING.Concert_Title@

```
db2 => SELECT TICKET.Concert_Code,CONCERT_SHOWING.Concert_Title,SUM(Discounted_Price) AS Total_Income
db2 (cont.) => FROM TICKET,CONCERT_SHOWING
db2 (cont.) => WHERE TICKET.Concert_Code=CONCERT_SHOWING.Concert_Code
db2 (cont.) => GROUP BY TICKET.Concert_Code,CONCERT_SHOWING.Concert_Title@

CONCERT_CODE CONCERT_TITLE       TOTAL_INCOME
------------ ------------------ ------------------------------
B            Beyond Tour                            560.00
G            GEM Tour                               900.00
J            Jay Chou Tour                         1780.00
JW           Jack World Tour                        700.00
W            Lee horm Tour                          500.00
WW           Wee World Tour                         480.00
```

Total concert held at each venue at different date

SELECT CONCERT_SHOWING.Venue_Code, Venue_Description, COUNT(Concert_Code) AS Total_Concert

FROM VENUE,CONCERT_SHOWING

WHERE VENUE.Venue_Code=CONCERT_SHOWING.Venue_Code

GROUP BY CONCERT_SHOWING.Venue_Code, VENUE.Venue_Description@

```
db2 => SELECT CONCERT_SHOWING.Venue_Code, Venue_Description, COUNT(Concert_Code) AS Total_Concert
db2 (cont.) => FROM VENUE,CONCERT_SHOWING
db2 (cont.) => WHERE VENUE.Venue_Code=CONCERT_SHOWING.Venue_Code
db2 (cont.) => GROUP BY CONCERT_SHOWING.Venue_Code, VENUE.Venue_Description@

VENUE_CODE VENUE_DESCRIPTION    TOTAL_CONCERT
---------- -------------------- -------------
BJ         KL Bukit Jalil                   3
G          Genting                          1
SM         Stadium Merdeka                  4
```

Number of adult purchase for the concert showing

SELECT TICKET.Concert_Code, Concert_Title, COUNT(Trans_No) AS Number_Of_Adult

FROM TICKET,CONCERT_SHOWING

WHERE TICKET.Concert_Code=CONCERT_SHOWING.Concert_Code

AND Ticket_Type='Adult'

GROUP BY TICKET.Concert_Code, CONCERT_SHOWING.Concert_Title@

```
db2 => SELECT TICKET.Concert_Code, Concert_Title, COUNT(Trans_No) AS Number_Of_Adult
db2 (cont.) => FROM TICKET,CONCERT_SHOWING
db2 (cont.) => WHERE TICKET.Concert_Code=CONCERT_SHOWING.Concert_Code
db2 (cont.) => AND Ticket_Type='Adult'
db2 (cont.) => GROUP BY TICKET.Concert_Code, CONCERT_SHOWING.Concert_Title@

CONCERT_CODE CONCERT_TITLE        NUMBER_OF_ADULT
------------ -------------------- ---------------
G            GEM Tour                           1
J            Jay Chou Tour                      2
JW           Jack World Tour                    1
W            Lee horm Tour                      1
```

Number of children purchase for the concert showing

SELECT TICKET.Concert_Code, Concert_Title, COUNT(Trans_No) AS Number_Of_Children

FROM TICKET,CONCERT_SHOWING

WHERE TICKET.Concert_Code=CONCERT_SHOWING.Concert_Code

AND Ticket_Type='Children'

GROUP BY TICKET.Concert_Code, CONCERT_SHOWING.Concert_Title@

```
db2 => SELECT TICKET.Concert_Code, Concert_Title, COUNT(Trans_No) AS Number_Of_Children
db2 (cont.) => FROM TICKET,CONCERT_SHOWING
db2 (cont.) => WHERE TICKET.Concert_Code=CONCERT_SHOWING.Concert_Code
db2 (cont.) => AND Ticket_Type='Children'
db2 (cont.) => GROUP BY TICKET.Concert_Code, CONCERT_SHOWING.Concert_Title@

CONCERT_CODE CONCERT_TITLE        NUMBER_OF_CHILDREN
------------ -------------------- ------------------
B            Beyond Tour                           2
G            GEM Tour                              1
J            Jay Chou Tour                         1
WW           Wee World Tour                        1
```

Total ticket that is sold on particular day

SELECT CONCERT_SHOWING.Concert_Code, CONCERT_SHOWING.Concert_Title, TICKET.Purchased_Date, COUNT(Trans_No) AS Total_Tickets

FROM TICKET,CONCERT_SHOWING

WHERE TICKET.Concert_Code=CONCERT_SHOWING.Concert_Code

GROUP BY CONCERT_SHOWING.Concert_Code,TICKET.Purchased_Date,CONCERT_SHOWING.Concert_Title

Order by TICKET.Purchased_Date@

```
db2 => SELECT CONCERT_SHOWING.Concert_Code, CONCERT_SHOWING.Concert_Title, TICKET.Purchased_Date, COUNT(Trans_No) AS Total_Tickets
db2 (cont.) => FROM TICKET,CONCERT_SHOWING
db2 (cont.) => WHERE TICKET.Concert_Code=CONCERT_SHOWING.Concert_Code
db2 (cont.) => GROUP BY CONCERT_SHOWING.Concert_Code,TICKET.Purchased_Date,CONCERT_SHOWING.Concert_Title
db2 (cont.) => Order by TICKET.Purchased_Date@

CONCERT_CODE CONCERT_TITLE       PURCHASED_DATE TOTAL_TICKETS
------------ ------------------- -------------- -------------
B            Beyond Tour         07/02/2017               2
G            GEM Tour            07/02/2017               2
J            Jay Chou Tour       07/02/2017               3
JW           Jack World Tour     07/02/2017               1
W            Lee horm Tour       07/02/2017               1
WW           Wee World Tour      07/02/2017               1
```

6.3 Triggers

Calculate price of the ticket

```
CREATE TRIGGER trg_CalcPrice
AFTER INSERT ON TICKET
REFERENCING NEW AS N
FOR EACH ROW mode db2sql
        UPDATE TICKET
        SET Discounted_Price=
        CASE
                WHEN Ticket_Type='Adult'
                        then (SELECT Price FROM AREA
                                WHERE Area.Area_Code=N.Area_Code)
                WHEN Ticket_Type='Children'
                        then (SELECT Price FROM AREA
                                WHERE Area.Area_Code=N.Area_Code) * 0.80
        END
        WHERE Trans_No=N.Trans_No@
```

```
db2 => call InsertTicket('T11','JW','2','V2','000003',CURRENT TIMESTAMP,CURRENT TIMESTAMP,'Children')@

  Return Status = 0
db2 => select * from ticket@

TRANS_NO CONCERT_CODE SEAT_NO AREA_CODE CUSTOMER_ID PURCHASED_DATE PURCHASED_TIME TICKET_TYPE         TICKET_REFUNDABLE    DISCOUNTED_PRICE
-------- ------------ ------- --------- ----------- -------------- -------------- ------------------ -------------------- ----------------
T1       J            1       V1        000001      02/07/2017     01:09:23       Adult              Available                     600.00
T2       J            2       V1        000001      02/07/2017     01:09:23       Children           Available                     480.00
T3       J            1       V2        000002      02/07/2017     01:09:23       Adult              Available                     700.00
T4       G            1       R1        000002      02/07/2017     01:09:23       Adult              Available                     500.00
T5       G            2       R1        000002      02/07/2017     01:09:23       Children           Available                     400.00
T6       W            1       R1        000003      02/07/2017     01:09:23       Adult              Available                     500.00
T7       B            3       R1        000004      02/07/2017     01:09:23       Children           Available                     400.00
T8       B            2       N         000005      02/07/2017     01:09:23       Children           Available                     160.00
T9       WW           1       V1        000006      02/07/2017     01:09:23       Children           Available                     480.00
T10      JW           1       V2        000003      02/07/2017     01:09:24       Adult              Available                     700.00
T11      JW           2       V2        000003      02/07/2017     01:36:41       Children           Available                     560.00
```

This trigger is used to calculate the price for Adult and Children. After every insert on TICKET table, the Discounted_Price column is auto updated to the original price for Adult and 80% of the original price for Children.

Delete ticket

```
CREATE TRIGGER trg_DropTicket
AFTER DELETE ON TICKET
REFERENCING OLD AS O
FOR EACH ROW MODE db2sql
BEGIN
IF ((SELECT COUNT(Customer_ID) FROM TICKET WHERE Customer_ID=O.Customer_ID) =0)
THEN
        DELETE FROM CUSTOMER
        WHERE Customer_ID = O.Customer_ID;
END IF ;
END@
```

```
db2 => select * from customer@

CUSTOMER_ID CUSTOMER_FIRSTNAME   CUSTOMER_LASTNAME    CUSTOMER_AGE CUSTOMER_GENDER CUSTOMER_PHONE CASHIER_ID LOCATION_ID
----------- -------------------- -------------------- ------------ --------------- -------------- ---------- -----------
000000      John                 Liew                           20 M               01225698745    C0001               1
000001      Jason                Loh                            13 M               0145986369     C0002               1
000002      Nicole               Teh                            20 F               032596554      C0002               1
000003      Jeff                 Liew                           20 M               01263549738    C0003               2
000005      Andrew               Teh                            20 F               038052395      C0008               5
000006      Yi gor               Teh                            20 M               038052328      C0008               5
```

```
db2 => select * from ticket@

TRANS_NO CONCERT_CODE SEAT_NO AREA_CODE CUSTOMER_ID PURCHASED_DATE PURCHASED_TIME TICKET_TYPE        TICKET_REFUNDABLE    DISCOUNTED_PRICE
-------- ------------ ------- --------- ----------- ------------- ------------- ----------------- -------------------- ----------------
T1       J            1       V1        000001      02/07/2017    01:09:23      Adult             Available                      600.00
T2       J            2       V1        000001      02/07/2017    01:09:23      Children          Available                      480.00
T3       J            1       V2        000002      02/07/2017    01:09:23      Adult             Available                      700.00
T4       G            1       R1        000002      02/07/2017    01:09:23      Adult             Available                      500.00
T5       G            2       R1        000002      02/07/2017    01:09:23      Children          Available                      400.00
T6       W            1       R1        000003      02/07/2017    01:09:23      Adult             Available                      500.00
T8       B            2       N         000005      02/07/2017    01:09:23      Children          Available                      160.00
T9       WW           1       V1        000006      02/07/2017    01:09:23      Children          Available                      480.00
T10      JW           1       V2        000003      02/07/2017    01:09:24      Adult             Available                      700.00
T11      JW           2       V2        000003      02/07/2017    01:36:41      Children          Available                      560.00
```

This trigger is used to remove the row from the CUSTOMER table after deleting from TICKET table. If the Customer_ID do not exist in the TICKET table after the delete, it will be automatically removed from the CUSTOMER table. If the Customer_ID exists, then the CUSTOMER table will remains the same. For example, the picture above originally have a Customer_ID which is '000004', 'T7' is the Trans_No for Customer_ID '000004', after the delete of Trans_No 'T7' from the TICKET table, the Customer_ID '000004' do not exist anymore in the TICKET table. Therefore, it is removed from the CUSTOMER table.

6.4 Stored Procedure

Insert ticket

```
CREATE PROCEDURE InsertTicket(IN Trans_No varchar(8), Code varchar(7), Seat_No
varchar(6), Area_Code varchar(3), Customer_ID varchar(6), Purchased_Date date,
Purchased_Time time, Ticket_Type varchar(20))
BEGIN
    IF (Purchased_Date <= (SELECT Concert_Date
        FROM CONCERT_SHOWING
        WHERE CONCERT_SHOWING.Concert_Code=Code)
        AND Purchased_Time <= (SELECT Concert_EndTime
                                FROM CONCERT_SHOWING
                                WHERE CONCERT_SHOWING.Concert_Code=Code )) THEN
          INSERT INTO TICKET (Trans_No,Concert_Code, Seat_No, Area_Code,
          Customer_ID,Purchased_Date,Purchased_Time , Ticket_Type) VALUES
          (Trans_No,Code, Seat_No, Area_Code, Customer_ID, Purchased_Date,
           Purchased_Time, Ticket_Type);
    ELSE
          SIGNAL SQLSTATE '78001'
          SET MESSAGE_TEXT='The concert you purchase is over. ';
END IF;
END@
```

```
db2 => call InsertTicket('T11','JW','2','V2','000003',CURRENT TIMESTAMP,CURRENT TIMESTAMP,'Children')@

  Return Status = 0
db2 => select * from ticket@

TRANS_NO CONCERT_CODE SEAT_NO AREA_CODE CUSTOMER_ID PURCHASED_DATE PURCHASED_TIME TICKET_TYPE        TICKET_REFUNDABLE    DISCOUNTED_PRICE
-------- ----------- ------- --------- ----------- ------------- ------------- ---------------- -------------------- ----------------
T1       J           1       V1        000001      02/07/2017    01:09:23      Adult            Available                      600.00
T2       J           2       V1        000001      02/07/2017    01:09:23      Children         Available                      480.00
T3       J           1       V2        000002      02/07/2017    01:09:23      Adult            Available                      700.00
T4       G           1       R1        000002      02/07/2017    01:09:23      Adult            Available                      500.00
T5       G           2       R1        000002      02/07/2017    01:09:23      Children         Available                      400.00
T6       W           1       R1        000003      02/07/2017    01:09:23      Adult            Available                      500.00
T7       B           3       R1        000004      02/07/2017    01:09:23      Children         Available                      400.00
T8       B           2       N         000005      02/07/2017    01:09:23      Children         Available                      160.00
T9       WW          1       V1        000006      02/07/2017    01:09:23      Children         Available                      480.00
T10      JW          1       V2        000003      02/07/2017    01:09:24      Adult            Available                      700.00
T11      JW          2       V2        000003      02/07/2017    01:36:41      Children         Available                      560.00
```

This procedure is to check whether the Purchased_Date is before or after the Concert_Date, if the Purchased_Date is before the Concert_Date then it will be possible to insert the values when this Procedure is called. If the Purchased_Date is after the Concert_Date, then it will send out a message text saying that 'The concert you purchase is over."

```
db2 => call InsertTicket('T12','JW','2','V2','000003','02/10/2017','01:00:00','Adult')@
SQL0438N  Application raised error or warning with diagnostic text: "The
concert you purchase is over. ".  SQLSTATE=78001
```

This is to prevent customers from buying those tickets which the concert is over.

Update the refundable of ticket

```
CREATE PROCEDURE UpdateRefund(IN Code varchar(7))
BEGIN
IF(DAY((SELECT Concert_Date
        FROM CONCERT_SHOWING
        WHERE CONCERT_SHOWING.Concert_Code=Code) –
        (SELECT current date FROM sysibm.sysdummy1)) < 7 ) THEN
         UPDATE TICKET
         SET Ticket_Refundable='Not available'
         WHERE TICKET.Concert_Code=Code;
ELSE
         UPDATE TICKET
         SET Ticket_Refundable='Available'
         WHERE TICKET.Concert_Code=Code;
END IF;
END@
```

```
db2 => call UpdateRefund('WW')@

  Return Status = 0
db2 => select * from ticket@

TRANS_NO CONCERT_CODE SEAT_NO AREA_CODE CUSTOMER_ID PURCHASED_DATE PURCHASED_TIME TICKET_TYPE        TICKET_REFUNDABLE   DISCOUNTED_PRICE
-------- ----------- ------- --------- ----------- ------------- ------------- ----------------- ------------------- ----------------
T1       J           1       U1        000001      02/07/2017    01:09:23      Adult             Available                       600.00
T2       J           2       U1        000001      02/07/2017    01:09:23      Children          Available                       480.00
T3       J           1       U2        000002      02/07/2017    01:09:23      Adult             Available                       700.00
T4       G           1       R1        000002      02/07/2017    01:09:23      Adult             Available                       500.00
T5       G           2       R1        000002      02/07/2017    01:09:23      Children          Available                       400.00
T6       W           1       R1        000003      02/07/2017    01:09:23      Adult             Available                       500.00
T8       B           2       N         000005      02/07/2017    01:09:23      Children          Available                       160.00
T9       WW          1       U1        000006      02/07/2017    01:09:23      Children          Not available                   480.00
T10      JW          1       U2        000003      02/07/2017    01:09:24      Adult             Available                       700.00
T11      JW          2       U2        000003      02/07/2017    01:36:41      Children          Available                       560.00
```

This procedure is to update the column Ticket_Refundable from the TICKET table. If the current date is more than 7 days away from the concert date, then it will change the value of the column Ticket_Refundable into ‘Available’. If the current date is less than 7 days away from the concert date, then it will changes the coloumn Ticket_Refundable into ’Not Available’.

6.5 View

Track ticket purchased location

```
CREATE VIEW TicketPurchasedLocation AS
        SELECT TICKET.Trans_No, CASHIER.Cashier_ID, CASHIER.Cashier_FirstName,
        CASHIER.Location_ID, SALES_LOCATION.Location_Description
        FROM TICKET, CUSTOMER, CASHIER, SALES_LOCATION
        WHERE TICKET.Customer_ID = CUSTOMER.Customer_ID
        AND CUSTOMER.Cashier_ID = CASHIER.Cashier_ID
        AND CASHIER.Location_ID = SALES_LOCATION.Location_ID@
```

```
db2 => select * from ticketPurchasedLocation@

TRANS_NO CASHIER_ID CASHIER_FIRSTNAME    LOCATION_ID LOCATION_DESCRIPTION
-------- ---------- -------------------- ----------- --------------------
T1       C0002      Michele                        1 IOI Mall
T10      C0003      Xyres                          2 TESCO
T2       C0002      Michele                        1 IOI Mall
T3       C0002      Michele                        1 IOI Mall
T4       C0002      Michele                        1 IOI Mall
T5       C0002      Michele                        1 IOI Mall
T6       C0003      Xyres                          2 TESCO
T7       C0004      Kevin                          2 TESCO
T8       C0008      Sean                           5 Mid Valley
T9       C0008      Sean                           5 Mid Valley
```

Track ticket belongs to which venue

```
CREATE VIEW Ticket_Venue AS
        SELECT Trans_No,Venue_Description
        FROM TICKET,CONCERT_SHOWING,VENUE
        WHERE TICKET.Concert_Code = CONCERT_SHOWING.Concert_Code AND
        CONCERT_SHOWING.Venue_Code = VENUE.Venue_Code@
```

```
TRANS_NO VENUE_DESCRIPTION
-------- --------------------
T1       KL Bukit Jalil
T10      KL Bukit Jalil
T2       KL Bukit Jalil
T3       KL Bukit Jalil
T4       Stadium Merdeka
T5       Stadium Merdeka
T6       Stadium Merdeka
T7       KL Bukit Jalil
T8       KL Bukit Jalil
T9       Stadium Merdeka
```

Track ticket belong to what seat

```
CREATE VIEW Ticket_Seat AS
        SELECT Trans_No, Seat_No, Area_Code
        FROM TICKET @
```

```
db2 => select * from ticket_seat@

TRANS_NO SEAT_NO AREA_CODE
-------- ------- ---------
T1       1       V1
T2       2       V1
T3       1       V2
T4       1       R1
T5       2       R1
T6       1       R1
T7       3       R1
T8       2       N
T9       1       V1
T10      1       V2
```

View concert venue

```
CREATE VIEW ConcertVenue_View AS
        SELECT Concert_Title, Concert_Date, Venue_Description
        FROM CONCERT_SHOWING, VENUE
        WHERE CONCERT_SHOWING.Venue_Code=Venue.Venue_Code@
```

```
db2 => select * from ConcertVenue_view@

CONCERT_TITLE        CONCERT_DATE VENUE_DESCRIPTION
-------------------- ------------ --------------------
Jay Chou Tour        27-02-2017   KL Bukit Jalil
Beyond Tour          26-02-2017   KL Bukit Jalil
GEM Tour             27-02-2017   Stadium Merdeka
Alice Tour           22-02-2017   Stadium Merdeka
Jeff Tan Tour        15-03-2017   Genting
Lee horm Tour        16-03-2017   Stadium Merdeka
Wee World Tour       07-02-2017   Stadium Merdeka
Jack World Tour      08-02-2017   KL Bukit Jalil
```

6.6 Subqueries or Nested Queries

Select concert that does not sell any tickets yet

```
SELECT Concert_Code , Concert_Title
FROM CONCERT_SHOWING
WHERE NOT EXISTS (SELECT Concert_Code
                  FROM Ticket
                  WHERE Ticket.Concert_Code = Concert_Showing.Concert_Code)@
```

```
db2 => SELECT Concert_Code , Concert_Title
db2 (cont.) => FROM CONCERT_SHOWING
db2 (cont.) => WHERE NOT EXISTS (SELECT Concert_Code
db2 (cont.) =>         FROM Ticket
db2 (cont.) =>         WHERE Ticket.Concert_Code = Concert_Showing.Concert_Code)@

CONCERT_CODE CONCERT_TITLE
------------ --------------------
A            Alice Tour
JT           Jeff Tan Tour
```

View concert that does not sell any tickets yet

```
SELECT Concert_Code , Concert_Title
FROM CONCERT_SHOWING
WHERE EXISTS (SELECT Concert_Code
              FROM Ticket
              WHERE Ticket.Concert_Code = Concert_Showing.Concert_Code)@
```

```
db2 => SELECT Concert_Code , Concert_Title FROM CONCERT_SHOWING
db2 (cont.) => WHERE EXISTS
db2 (cont.) => (SELECT Concert_Code FROM Ticket
db2 (cont.) => WHERE   Ticket.Concert_Code = Concert_Showing.Concert_Code)@

CONCERT_CODE CONCERT_TITLE
------------ --------------------
J            Jay Chou Tour
B            Beyond Tour
G            GEM Tour
W            Lee horm Tour
WW           Wee World Tour
JW           Jack World Tour
```

6.7 New Queries Not Covered In Lecture

Receive date from the Operating System

```
CREATE PROCEDURE UpdateRefund(IN Code varchar(7))
BEGIN
IF(DAY((SELECT Concert_Date FROM CONCERT_SHOWING WHERE
        CONCERT_SHOWING.Concert_Code=Code)
        -(SELECT current date FROM sysibm.sysdummy1)) < 7 ) THEN
        UPDATE TICKET
        SET Ticket_Refundable='Not available'
        WHERE TICKET.Concert_Code=Code;
ELSE
        UPDATE TICKET
        SET Ticket_Refundable='Available'
        WHERE TICKET.Concert_Code=Code;
END IF;
END@
```

After UpdateRefund('JW') is called:

```
db2 => select * from ticket@

TRANS_NO CONCERT_CODE SEAT_NO AREA_CODE CUSTOMER_ID PURCHASED_DATE PURCHASED_TIME TICKET_TYPE        TICKET_REFUNDABLE    DISCOUNTED_PRICE
-------- ------------ ------- --------- ----------- ------------- ------------- ------------------ -------------------- --------------
T1       J            1       V1        000001      07-02-2017    04:12:26      Adult              Available                     600.00
T2       J            2       V1        000001      07-02-2017    04:12:26      Children           Available                     480.00
T3       J            1       V2        000002      07-02-2017    04:12:26      Adult              Available                     700.00
T4       G            1       R1        000002      07-02-2017    04:12:26      Adult              Available                     500.00
T5       G            2       R1        000002      07-02-2017    04:12:26      Children           Available                     400.00
T6       W            1       R1        000003      07-02-2017    04:12:26      Adult              Available                     500.00
T7       B            3       R1        000004      07-02-2017    04:12:26      Children           Available                     400.00
T8       B            2       N         000005      07-02-2017    04:12:26      Children           Available                     160.00
T9       WW           1       V1        000006      07-02-2017    04:12:26      Children           Available                     480.00
T10      JW           1       V2        000003      07-02-2017    04:12:27      Adult              Not available                 700.00
```

This is because the concert date for Jack World Tour is on 08/02/2017 while currently the system date is 07/02/2017. Therefore, customer cannot refund their ticket before 7 days of the concert.

Auto Increment (Generated New Number for New Row)

```
CREATE TABLE SALES_LOCATION(
        Location_ID integer NOT NULL GENERATED ALWAYS AS IDENTITY,
        Location_Description varchar(20) NOT NULL,
        Location_City varchar(20),
        Location_State  varchar(20),
        Location_PostCode varchar(5) ,
        PRIMARY KEY(Location_ID),
        CONSTRAINT Location_Unq UNIQUE(Location_Description)
);
```

After insertion operation executed:

```
insert into sales_location (Location_Description,Location_City,Location_State,Location_PostCode)
Values ('Giant','Puchong','Selangor','47100')@
```

```
db2 => select * from sales_location@

LOCATION_ID LOCATION_DESCRIPTION LOCATION_CITY       LOCATION_STATE      LOCATION_POSTCODE
----------- ------------------- ------------------- ------------------- ---------------
          1 Setia Walk          Puchong             Selangor            47100
          2 Giant               Puchong             Selangor            47100
```

Fetch Top 5 Best Performance among the Cashier

```
SELECT    Customer.Cashier_ID ,Cashier_FirstName,Cashier_LastName,COUNT(Customer.Cashier_ID)
AS CustomerServed FROM CUSTOMER,Cashier
WHERE Customer.Cashier_ID = Cashier.Cashier_ID
GROUP BY Customer.Cashier_ID,Cashier_FirstName,Cashier_LastName
ORDER BY COUNT(Customer.Cashier_ID) DESC
FETCH FIRST 5 ROWS ONLY @
```

```
CASHIER_ID CASHIER_FIRSTNAME    CASHIER_LASTNAME     CUSTOMERSERVED
---------- -------------------- -------------------- --------------
C0008      Sean                 Chow                            2
C0002      Michele              Hee                             2
C0005      Selina               Ho                              1
C0001      Justin               Ng                              1
C0004      Kevin                Liew                            1
```