

## Introduction:

This database schema portrays a super shop management system where general people are used to buy their daily useful and essential necessary regular household products. The shopkeeper is really spent their valuable time in giving product to their customers and taking money from purchasers. That's why, to save his valuable time and to earn more profit, he has to need a digital shop management technology which can be solved easily by database. Through database, he can easily calculate his profitable money, expance, item, sale details and store them a lot of time. Thus, he can save his valuable time and use his extra time in more effective tasks. Hence, he can give a daily report in his central market. Therefore, database is a part and parcel of daily shop management system.

## Problem Statement:

### Q1. Database schema:

Customer (Cust\_id: integer, cust\_name:string)  
relation\_n\_i (cust\_id, item\_id, bill\_no)  
Item (item\_id: integer, item\_name: string, price:  
Sale (bill\_no: integer, bill\_date:date, qty\_sold:integer)

For the above schema, perform the following-

- i. Create the tables with the appropriate integrity constraints, insert around 10 records in each of the tables.
- ii. Design the relationship set table to associate the relationship among the given entity set.
- iii. Given count of how many items have been purchased by each customer in different bill number.
- iv. List the total bill amount with the quantity sold, total price of the items.
- v. Determine the total sold of a particular item

in the month July 2021.

vi. Produce 20% less bill for a customer who have bill more than BDT 3000.

Designed database (i.e., Relational or Entity Relationship model):

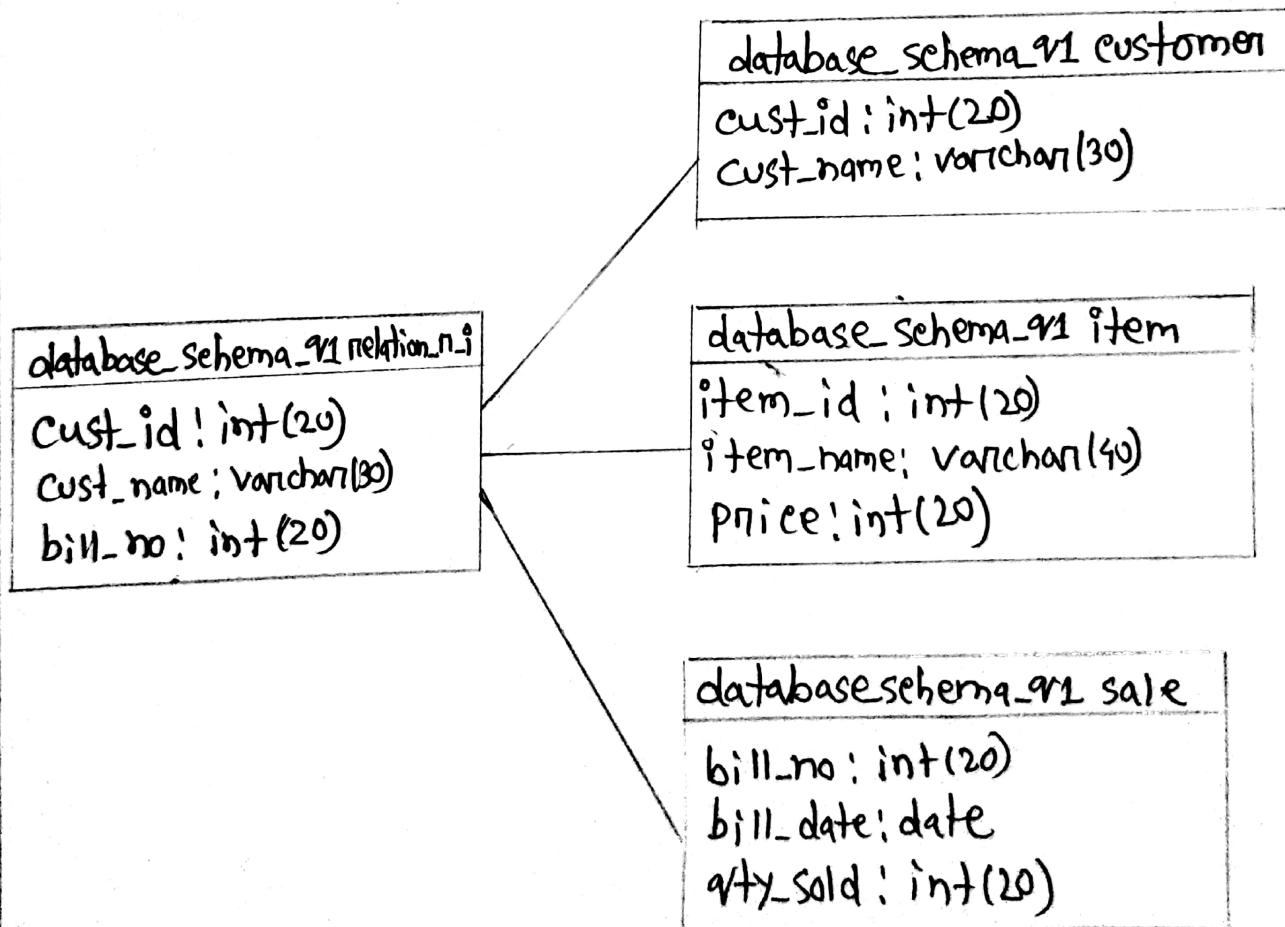


Figure 1.1 : Entity Relationship Model.

Submitted query at the Query processor  
with results:

- i. Create the tables with the appropriate integrity constraints, insert around 10 records in each of the tables.

First creating the database,

`CREATE DATABASE database-schema-91;`

Then creating the tables,

Customer:

```
CREATE TABLE customer(  
    cust_id int(20) NOT NULL,  
    cust_name varchar(30),  
    PRIMARY KEY (cust_id));
```

Item:

```
CREATE TABLE item(  
    item_id int(20) NOT NULL,  
    item_name varchar(40),  
    price int(20),  
    primary key (item_id));
```

Sale:

```
CREATE TABLE Sale (
    bill_no int(20) NOT NULL,
    bill_date date,
    qty_sold int(20),
    PRIMARY KEY (bill_no));
```

relation-n-i;

```
CREATE TABLE relation-n-i (
    cust_id int,
    FOREIGN KEY (cust_id) REFERENCES customer(cust_id),
    item_id int,
    FOREIGN KEY (item_id) REFERENCES item(item_id),
    bill_no int,
    FOREIGN KEY (bill_no) REFERENCES Sale(bill_no);
```

Then Insert the value in the respective Table,

Customer:

INSERT INTO customer ( cust\_id , cust\_name )

VALUES ( 101 , " Tamim " ) ,

( 102 , " Soumya " ) ,

( 103 , " Liton " ) ,

( 104 , " Mahmudullah " ) ,

( 105 , " Musfiq " ) ,

( 106 , " Sakib " ) ,

( 107 , " Sohan " ) ,

( 108 , " Mahedi " ) ,

( 109 , " Mustafizur " ) ,

( 110 , " Nasum " ) ;

Output:

cust_id	cust_name
101	Tamim
102	Soumya
103	Liton
104	Mahmudullah
105	Musfiq
106	Sakib
107	Sohan
108	Mahedi
109	Mustafizur
110	Nasum

Item:

```
INSERT INTO item(item_id, item_name, price)
VALUES (10001, "icecream", 300),
       (10002, "sugar", 50),
       (10003, "chips", 100),
       (10004, "pen", 10),
       (10005, "cookies", 200),
       (10006, "chocklate", 200),
       (10007, "biscuit", 200),
       (10008, "nuts", 200),
       (10009, "juice", 70),
       (10010, "tea", 15);
```

Output:

item_id	item_name	price
10001	icecream	300
10002	sugar	50
10003	chips	100
10004	pen	10
10005	cookies	200
10006	chocklate	200
10007	biscuit	200
10008	nuts	200
10009	juice	70
10010	tea	15

Sale :

INSERT INTO Sale (bill-no, bill-date, qty-sold)  
VALUES (1, "2021-07-01", 15),  
(2, "2021-07-01", 20),  
(3, "2021-07-02", 20),  
(4, "2021-07-03", 12),  
(5, "2021-07-04", 5),  
(6, "2021-07-04", 8),  
(7, "2021-07-04", 25),  
(8, "2021-07-05", 28),  
(9, "2021-07-06", 32),  
(10, "2021-07-07", 35);

Output:

bill-no	bill-date	qty-sold
1	2021-07-01	15
2	2021-07-01	20
3	2021-07-02	20
4	2021-07-03	12
5	2021-07-04	5
6	2021-07-04	8
7	2021-07-04	25
8	2021-07-05	28
9	2021-07-06	32
10	2021-07-07	35

relation\_n\_i :

INSERT INTO relation\_n\_i (cust\_id, item\_id, bill\_no)

VALUES (101, 10001, 5),  
(103, 10002, 8),  
(102, 10003, 1),  
(105, 10004, 3),  
(104, 10005, 4),  
(107, 10006, 7),  
(106, 10007, 9),  
(109, 10008, 2),  
(108, 10009, 10),  
(110, 10010, 6);

Output:

cust_id	item_id	bill_no
101	10001	5
102	10002	8
103	10003	1
104	10004	3
105	10005	4
106	10006	7
107	10007	9
108	10008	2
109	10009	10
110	10010	6

(ii) Design the relationship set table to associate among the given entity set.

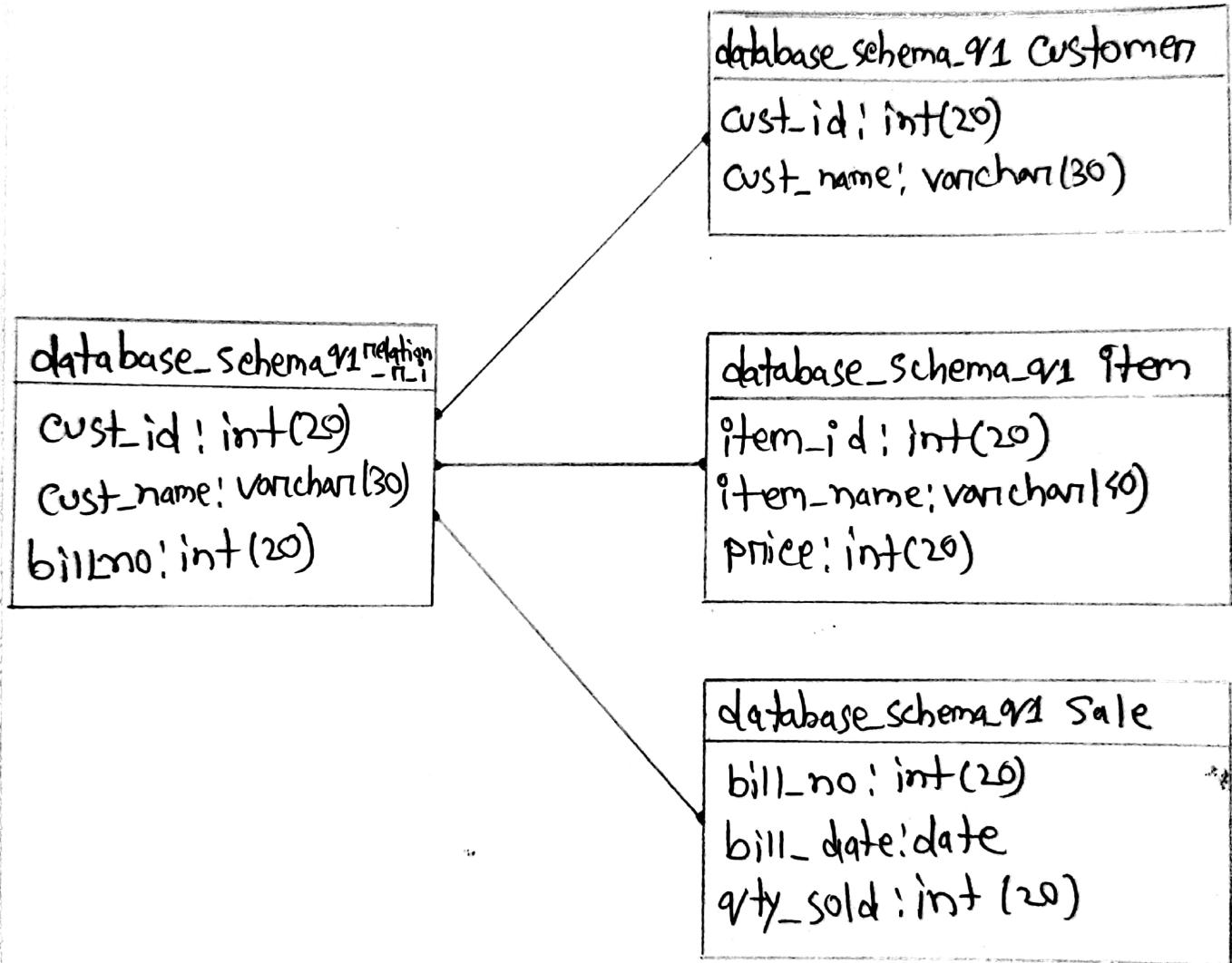


Figure 1.2 : Relationship Set Table to associate among the given entity set.

(iii) Give a count of how many items have been purchased by each customer in different bill number.

SQL operation:

Select cust\_id, qty\_sold, bill\_no From relation\_n-i JOIN sale USING (bill\_no);

Output:

cust_id	qty_sold	bill_no
101	5	5
103	28	8
102	15	1
105	20	3
104	12	9
107	25	7
106	32	9
109	20	2
108	35	10
110	8	6

(iv) List the total bill amount with the quantity sold, total price of the items.

SQL operation:

Select bill-no, item-id, qty-sold, (price\*qty sold)  
 AS total-price FROM item JOIN (SELECT bill-no,  
 item-id, qty-sold FROM relation-n-i JOIN  
 sale USING (bill-no)) AS list-of-total-price  
 USING (item-id);

Output:

bill-no	item-id	qty_sold	total-price
5	10001	5	1500
8	10002	28	1900
1	10003	15	1500
3	10004	20	200
4	10005	12	2400
7	10006	25	10000
9	10007	32	6400
2	10008	20	9000
10	10009	35	2950
6	10010	8	120

(V) Determine the total sold of a particular item in the month July 2021.

SQL operation:

```
Select item_id, SUM(qty_sold)
AS TOTAL_Quantity_OF_Sold FROM
relation_n_i JOIN Sale USING(bill_no)
WHERE bill_date BETWEEN '2021-07-01'
AND '2021-07-31' GROUP BY (item_id);
```

Output:

item_id	TOTAL_Quantity_of_Sold
10001	5
10002	28
10003	15
10004	20
10005	12
10006	25
10007	32
10008	20
10009	35
10010	8

(vi) Produce 20% less bill for a customer who have bill more than BDT 3000,

SQL operation:

```

SELECT bill_no, item_id, qty_sold,
       (price * qty_sold) AS TOTAL_PRICE,
       (price * qty_sold * .8) AS DISCOUNTED_PRICE
  FROM item JOIN (SELECT bill_no, item_id,
                        qty_sold FROM relation_n_i JOIN sale
                       USING (bill_no)) AS DISCOUNTED_TABLE
             USING (item_id)
 WHERE total_price >= 3000;
    
```

Output:

bill_no	item_id	qty_sold	TOTAL_PRICE	DISCOUNTED_PRICE
7	1000625	2510000	10000	8000
9	1000732	326400	6900	5120
2	1000820	205000	9000	3200

## Conclusion:

This database schema provides a preliminary knowledge about primary key, foreign key, subquery, natural join etc. It ~~of~~ assists how to connect the table into a new relation set relation - n-i. Here we can watch the entity relationship model where we see there's a many to many relationship. We can teach that 'where' clause is conditioned for tuples whereas 'Having' clause is conditioned on groups. Here we works on a shop management system than how to make a shop digitalized using database management system.

## Introduction of Design Town:

'Town' is a solution for general people in the aspect of journey. It is a very effective system for reducing pressure. It will work on various sections. Through this system we can easily install buses (if one is owner), book a ticket for one place to another place, even one can cancel ticket and can be able to show ticket easily. Here we develop a ticket booking system for demonstrating how to use other procedural languages like SQL in database system. It is in business with several buses connecting different places country.

Problem Statement:

Q2. Design Tourn

Analyze the problem and design the relations in it using the guided constraints. Identify what data has to be persisted in the databases.

The following are the entities:

Bus (BUS\_NO NOT NULL INTEGER(5), SOURCE  
VARCHAR(20), DESTINATION VARCHAR(20),  
COUCH\_TYPE VARCHAR(10), FAIR NUMBER(3))

Reservation (PNR\_NO NUMBER(9), NO\_OF\_SEATS  
NUMBER(8), ADDRESS VARCHAR(50), CONTACT\_NO  
NUMBER(10), STATUS CHAR(3))

Ticket (TICKET\_NO NOT NULL NUMBER(9),  
AGE NUMBER(4), SEX NOT NULL CHAR(1),  
SOURCE VARCHAR(2), DESTINATION VARCHAR(20),  
DEP\_TIME VARCHAR(4), JOURNEY\_DATE (DATE))

Passenger (TICKET\_NO NOT NULL NUMBER(9),  
NAME VARCHAR(40), AGE NUMBER(4), SEX NOT NULL)

CHAR(4), CONTACT\_NO NUMBER (10), FOREIGN KEY TICKET\_NO)

Cancellation (PNR\_NO NUMBER (9), NO\_OF\_SEATS NUMBER (8), ADDRESS VARCHAR (50), CONTACT\_NO NUMBER (10), STATUS CHAR(3), FOREIGN KEY PNR\_NO, CONTACT\_NO)

Now answer the following queries using

SQL:

1. How many passenger booked ticket for bus no 22 which is scheduled to run from Rajshahi to Dhaka on 1<sup>st</sup> of July 21, at 12:00PM
2. How many tickets have been canceled in the month of July 2021.
3. List the details of the customers who are travelling under age of 18 in bus no 22 on 1<sup>st</sup> July 2021 at 12:00PM

Designed database (i.e Relational on Entity Relationship model):

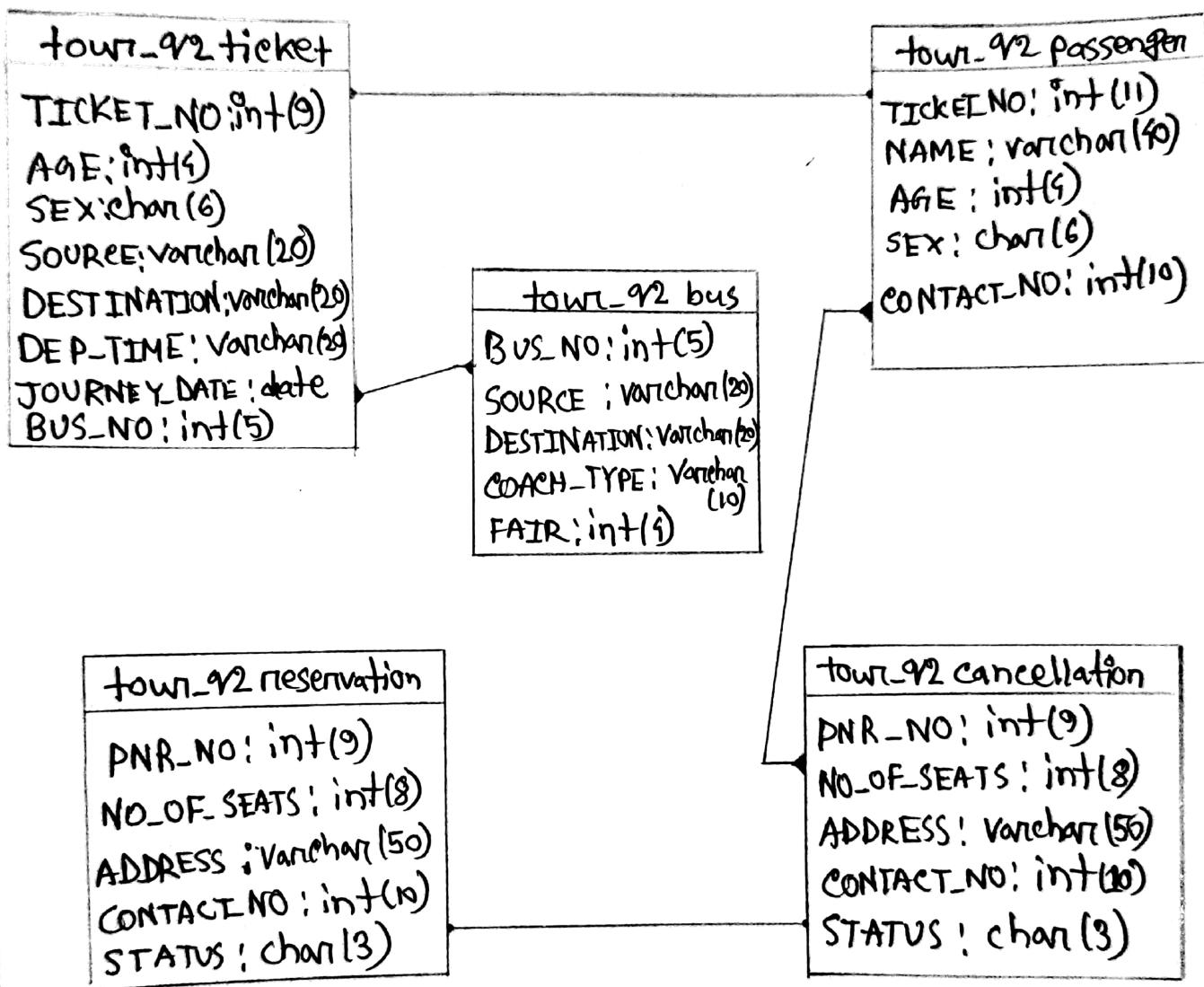


Figure 2.1 : Entity Relationship Model.

First creating the database,

CREATE DATABASE TOUR-Q2;

Then creating the tables,

Bus:

CREATE TABLE BUS(  
BUS\_NO int(5) NOT NULL,  
SOURCE varchar(20),  
DESTINATION varchar(20),  
COACH\_TYPE varchar(10),  
FAIR int(4),  
PRIMARY KEY (BUS\_NO));

Reservation:

CREATE TABLE Reservation(  
PNR\_NO int(9),  
NO\_OF\_SEATS int(8),  
ADDRESS varchar(50),  
CONTACT\_NO int(10),  
STATUS char(3),  
PRIMARY KEY (PNR\_NO));

## Ticket:

```
CREATE TABLE Ticket (
    TICKET_NO int(9) NOT NULL,
    AGE int(4),
    SEX char(6) NOT NULL,
    SOURCE varchar(20),
    DESTINATION varchar(20),
    DEP_TIME varchar(20),
    JOURNEY_DATE date,
    PRIMARY KEY (TICKET_NO),
    BUS_NO int(5) NOT NULL,
    FOREIGN KEY (BUS_NO) REFERENCES
    ticket (TICKET_NO);
    bus (BUS_NO));
```

## Passenger:

```
CREATE TABLE Passenger (
    TICKET_NO int(9) NOT NULL,
    NAME varchar(40),
    AGE int(4),
    SEX char(6) NOT NULL,
```

CONTACT-NO int(10),  
PRIMARY KEY (CONTACT-NO),  
FOREIGN KEY (TICKET-NO)  
REFERENCES ticket (TICKET-NO);

Cancellation:

CREATE TABLE cancellation(  
PNR\_NO int(9),  
NO\_OF\_SEATS int(8),  
ADDRESS varchar(50),  
CONTACT\_NO int,  
STATUS char(3)  
FOREIGN KEY (PNR\_NO) REFERENCES  
reservation (PNR\_NO),  
FOREIGN KEY (CONTACT\_NO) REFERENCES  
passenger (CONTACT\_NO));

Then Insert the value in the respective Table,

Bus:

INSERT INTO bus

( BUS\_NO, SOURCE, DESTINATION, COACH\_TYPE,  
 FAIR) VALUES ('20', 'Dhaka', 'Rajshahi', 'AC', 1600),  
 ('22', 'Rajshahi', 'Dhaka', 'AC', 1600),  
 ('30', 'Dhaka', 'Khulna', 'Non AC', 800),  
 ('35', 'Chittagong', 'Dhaka', 'AC', 2000),  
 ('40', 'Dhaka', 'Kolkata', 'AC', 5300);

Output:

BUS_NO	SOURCE	DESTINATION	COACH_TYPE	FAIR
20	Dhaka	Rajshahi	AC	1600
22	Rajshahi	Dhaka	AC	1600
30	Dhaka	Khulna	Non AC	800
35	Chittagong	Dhaka	AC	2000
40	Dhaka	Kolkata	AC	5300

## Reservation:

```
INSERT INTO reservation
(PNR_NO, NO_OF_SEATS, ADDRESS, CONTACT_NO, STATUS)
VALUES ('1', '5', '1216 Minpur, Dhaka', 01812375570,
       'booked'),
       ('2', '1', '235, Vodra, Rajshahi', 01813050250,
       'booked'),
       ('3', '4', 'Modonhata, Gulistan, 1212', 01675641332,
       'booked'),
       ('4', '2', 'Kamalpur, Chittagong, 1217', 01813050251,
       'booked'),
       ('5', '1', 'Rupnagar, Minpur, 1215', 01812376560,
       'booked');
```

## Output:

PNR_NO	NO_OF_SEATS	ADDRESS	CONTACT_NO	STATUS
1	5	1216 Minpur, Dhaka	01812375570	booked
2	1	235, Vodra, Rajshahi	01813050250	booked
3	4	Modonhata, Gulistan, 1212	01675641332	booked
4	2	Kamalpur, Chittagong 1217	01813050251	booked
5	1	Rupnagar, Minpur, 1215	01812376560	booked

Ticket:

INSERT INTO ticket  
 (TICKET\_NO, AGE, SEX, SOURCE, DESTINATION, DEP\_TIME,  
 JOURNEY\_DATE, BUS\_NO) VALUES  
 ('301', '25', 'Male', 'Dhaka', 'Rajshahi', '12:00PM',  
 '2021-07-01', '20'),  
 ('302', '17', 'Male', 'Rajshahi', 'Dhaka', '12:00PM',  
 '2021-07-01', '22'),  
 ('303', '45', 'Female', 'Dhaka', 'Khulna', '11:00PM',  
 '2021-07-01', '30'),  
 ('304', '55', 'Female', 'Chittagong', 'Dhaka', '12:00PM',  
 '2021-07-01', '35'),  
 ('305', '65', 'Male', 'Dhaka', 'Kolkata', '11:00PM',  
 '2021-07-01', '40');

Output:

TICKET_NO	AGE	SEX	SOURCE	DESTINATION	DEP_TIME	JOURNEY_DATE	BUS_NO
301	25	Male	Dhaka	Rajshahi	12:00PM	2021-07-01	20
302	17	Male	Rajshahi	Dhaka	12:00PM	2021-07-01	22
303	45	Female	Dhaka	Khulna	11:00PM	2021-07-01	30
304	55	Female	Chittagong	Dhaka	12:00PM	2021-07-01	35
305	65	Male	Dhaka	Kolkata	11:00PM	2021-07-01	40

Passenger:

INSERT INTO passenger

(TICKET\_NO, NAME, AGE, SEX, CONTACT\_NO) VALUES  
 ('301', 'Tamim', 25, 'Male', 01812375570),  
 ('302', 'Liton', 17, 'Male', 01813050250),  
 ('303', 'Shisn', 45, 'Female', 01675691332),  
 ('304', 'Fahmida', 55, 'Female', 01813050251),  
 ('305', 'Sakib', 65, 'Male', 01812376560);

Output:

TICKET_NO	NAME	AGE	SEX	CONTACT_NO
301	Tamim	25	Male	01812375570
302	Liton	17	Male	01813050250
303	Shisn	45	Female	01675691332
304	Fahmida	55	Female	01813050251
305	Sakib	65	Male	01812376560

Cancellation:

```
INSERT INTO cancellation(
    PNR_NO, NO_OF_SEATS, ADDRESS, CONTACT_NO, STATUS)
VALUES ('3', '4', 'Modonbata, Gulistan, 1212',
        01675691332, 'cancel'),
       ('5', '1', 'Rupnagar, Mimpuri, 1215',
        01812376560, 'cancel');
```

Output:

PNR_NO	NO_OF_SEATS	ADDRESS	CONTACT_NO	STATUS
3	4	Modonbata, Gulistan, 1212	01675691332	cancel
5	1	Rupnagar, Mimpuri, 1215	01812376560	cancel

Submitted query at the Query Processor with Results:

(i) How many passenger booked ticket for bus no 22 which is scheduled to run from Rajshahi to Dhaka on 1st of July 21, at 12:00 PM.

SQL Operation:

```
Select COUNT(TICKET_NO) as TOTAL_BOOKED_TICKET  
in Month of July FROM ticket WHERE BUS_NO = 22 AND  
JOURNEY_DATE = '2021-07-01' AND DEPTIME = '12:00PM'  
GROUP BY BUS_NO;
```

Output:

Canceled_Ticket_In_Month_of_July
2

TOTAL_BOOKED_TICKET
1

(ii) How many tickets have been canceled in the month of July 2021?

SQL operation:

Select count (TICKET-NO) as canceled-Ticket  
- In - Month - Of - July From ticket where  
Ticket TICKET-NO IN (Select TICKET-NO  
From passenger where CONTACT-NO IN  
(Select CONTACT-NO From cancellation  
JOIN reservation USING (PNR-NO,~~NAME~~,  
CONTACT-NO))) AND JOURNEY-DATE BETWEEN  
'2021-07-01' AND '2021-07-31';

Output:

Canceled_Ticket_In_Month_of_July
2

(iii) List the details of the customer who are travelling under age of 18 in bus no 22 on 1st July 2021 at 12:00PM

SQL Operation:

Select TICKET\_NO, BUS\_NO, NAME, AGE, SEX, SOURCE, DESTINATION, DEP\_TIME, JOURNEY\_DATE from ticket INNER JOIN passenger USING (TICKET\_NO, AGE, SEX)  
 WHERE BUS\_NO = 22 AND JOURNEY\_DATE = '2021-07-01' AND DEP\_TIME = '12:00PM'  
 AND AGE < 18;

Output:

TICKET_NO	BUS_NO	NAME	AGE	SEX	SOURCE	DESTINA-TION	DEP-TIME	JOURNEY-DATE
302	22	Liton	17	Male	Rajshahi	Dhaka	12:00 PM	2021-07-01

## Conclusion:

This database schema is quite complex, difficult and challenging. Sometimes XAMPP and localhost needs some clarification. The logical level of this database query is handled by the collaboration of human brain. This database schema provides an intermediate knowledge about inner join, subquery, primary key, foreign key. It assists how to use the connection between ~~the~~ relationship of two table. The relationship model shows that it's a many to many relationship here we works on a travel management system which represents how to digital the travel management system using database.