A black and white sign

Description automatically generated with low confidence

**DATABASES**

October 2021 -- February 2022

**SCHOOL OF INFOCOMM TECHNOLOGY**

Diploma in Information Technology

Diploma in Data Science

Diploma in Cyber Security & Digital Forensic

**(50 % of DB Module)**

**10 Jan 2022 – 29 Jan 2022 (3 Weeks)**

|  |  |  |
| --- | --- | --- |
| Deliverables | Deadline | Submission |
| Softcopy | Saturday 29 Jan 2022, 23:59 pm | MEL Assignment Submission link |
| Hardcopy | Monday 31 Jan 2022, 10:00 am | Blk 31 Level 8 (Admin Office),  Pigeon-holes (Databases) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| DB Class Group | | | T08 | |
| Team Number | | | 13 | Team Grade: |
| Tutor | | |  | |
| Members | Student No. | Student Name | | Individual Grade |
|  | Ngui Jia Le Sherlena | |  |
|  | Vincent Tan Guan Shen | |  |

Contents

[ER Model - 3 -](#_Toc94354663)

[Additional Assumptions - 4 -](#_Toc94354664)

[Constraints - 4 -](#_Toc94354665)

[Mapped Relations - 5 -](#_Toc94354666)

[Data Dictionary - 6 -](#_Toc94354667)

[SQL Commands - 15 -](#_Toc94354668)

[SQL Commands for Tables Creation - 15 -](#_Toc94354669)

[SQL Commands for Insertion of Data - 22 -](#_Toc94354670)

[SQL output for Passenger Table - 47 -](#_Toc94354671)

[SQL output for ContactNumber Table - 48 -](#_Toc94354672)

[SQL output for Reservation Table - 51 -](#_Toc94354673)

[SQL output for Booking Table - 53 -](#_Toc94354674)

[SQL output for Event Table - 54 -](#_Toc94354675)

[SQL output for EventType Table - 54 -](#_Toc94354676)

[SQL output for EventSession Table - 55 -](#_Toc94354677)

[SQL output for Eatery Table - 56 -](#_Toc94354678)

[SQL output for Dish Table - 57 -](#_Toc94354679)

[SQL output for Ingredient Table - 58 -](#_Toc94354680)

[SQL output for Contain Table - 60 -](#_Toc94354681)

[SQL output for Cuisine Table - 65 -](#_Toc94354682)

[SQL output for FoodCategory Table - 65 -](#_Toc94354683)

[SQL output for Categorised\_in Table - 66 -](#_Toc94354684)

[SQL output for CSDish Table - 69 -](#_Toc94354685)

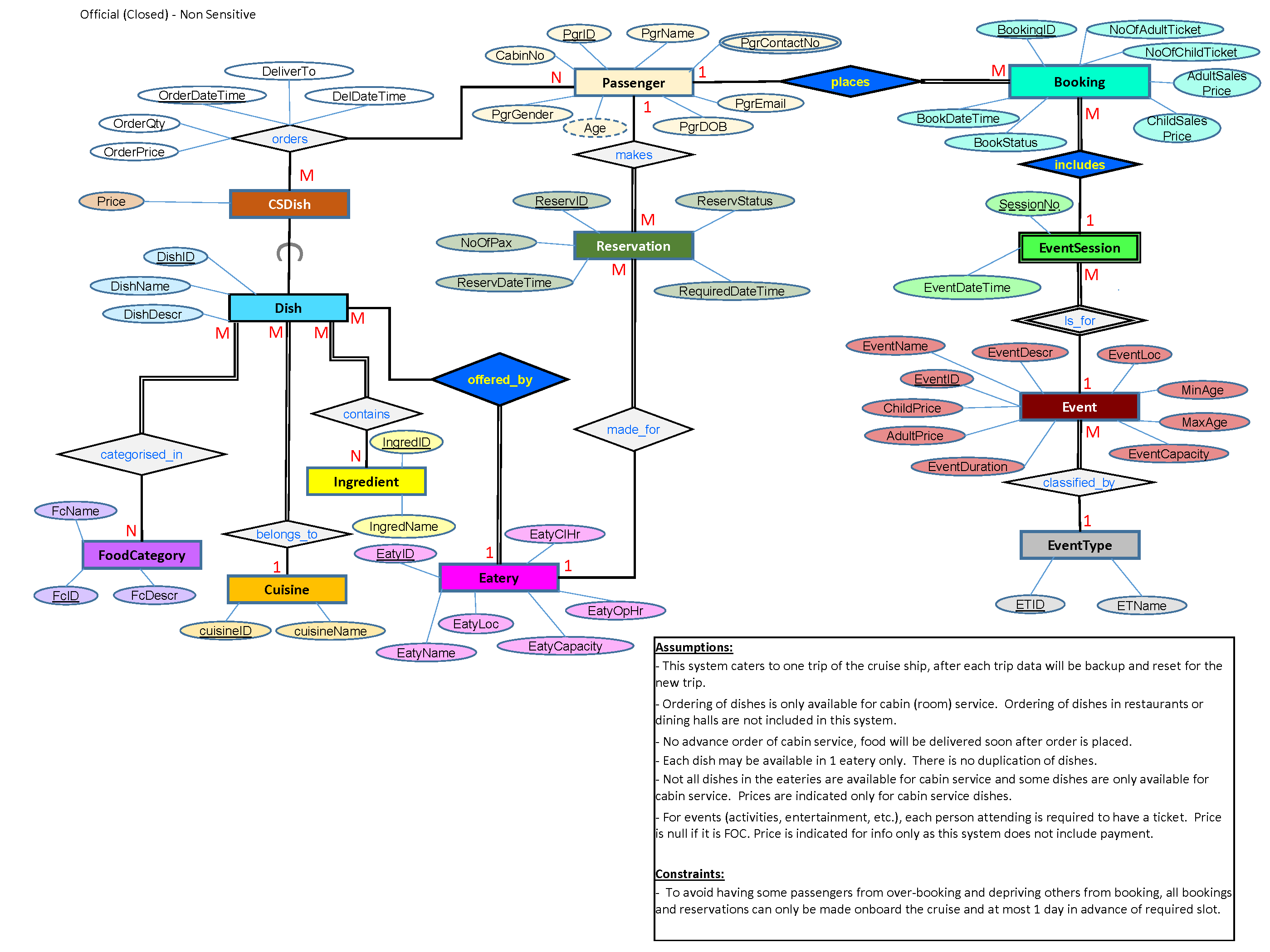
[SQL output for Orders Table - 70 -](#_Toc94354686)

[List of Queries - 72 -](#_Toc94354687)

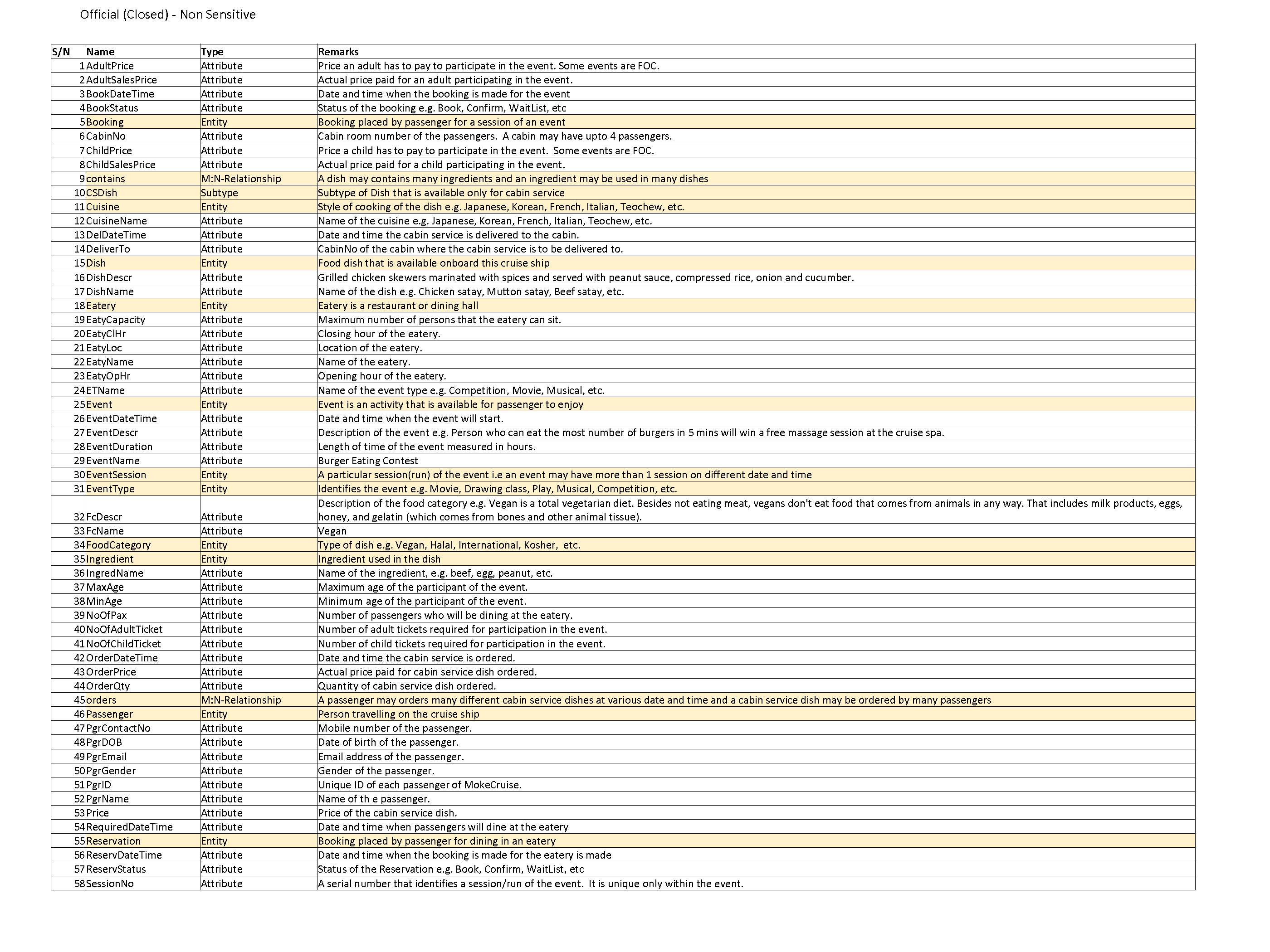
[Sherlena’s Queries - 72 -](#_Toc94354688)

[Vincent’s Queries - 75 -](#_Toc94354689)

# ER Model







# Additional Assumptions

1. There can be eateries that have yet to be built but has a name.
2. All passengers are required to have unique contact numbers.
3. All passengers are required to have an email address.
4. BookStatus and ReservStatus values are set to Booked, Confirmed, and Wait List where Booked is the default status.
5. DelDateTime is set to NULL when the cabin service has not been delivered to the cabin.
6. Before the passenger data is entered, the number of passengers staying in the room is confirmed to be four or below.
7. The value entered for the DeliverTo in Orders has already been checked to be a valid CabinNo.
8. Only events bookable at the point of time will be shown to passengers for them to make the booking.
9. In order for passengers to purchase a child ticket the passenger must input the PgrID for the child which will check if they are under 12.
10. This system caters to one trip of the cruise ship, after each trip data will be backup and reset for the new trip.
11. Ordering of dishes is only available for cabin (room) service. Ordering of dishes in restaurants or dining halls are not included in this system.
12. No advance order of cabin service, food will be delivered soon after order is placed.
13. Each dish may be available in 1 eatery only. There is no duplication of dishes.
14. Not all dishes in the eateries are available for cabin service and some dishes are only available for cabin service. Prices are indicated only for cabin service dishes.
15. For events (activities, entertainment, etc.), each person attending is required to have a ticket. Price is null if it is FOC. Price is indicated for info only as this system does not include payment.

# Constraints

1. To avoid having some passengers from over-booking and depriving others from booking, all bookings and reservations can only be made onboard the cruise and at most 1 day in advance of required slot.

# Mapped Relations

Passenger (PgrID, PgrName, PgrGender, PgrEmail, PgrDOB, CabinNo)

ContactNumber (<PgrID>, PgrContactNo)

Reservation (ReservID, ReservDateTime, RequiredDateTime, NoOfPax, ReservStatus, <PgrID>, <EatyID>)

Booking (BookingID, NoOfAdultTicket, AdultSalesPrice, NoOfChildTicket, ChildSalesPrice, BookDateTime, BookStatus, <EventID, SessionNo>, <PgrID>)

Event (EventID, EventName, EventDescr, EventCapacity, EventDuration, EventLoc, MinAge, MaxAge, AdultPrice, ChildPrice, <ETID>)

EventType (ETID, ETName)

EventSession (<EventID>, SessionNo, EventDateTime)

Eatery (EatyID, EatyName, EatyClHr, EatyOpHr, EatyLoc, EatyCapacity)

Ingredient (IngredID, IngredName)

Contain (<DishID>, <IngredID>)

Cuisine (CuisineID, CuisineName)

Dish (DishID, DishName, DishDescr, <EatyID>, CuisineID)

FoodCategory (FcID, FcName, FcDescr)

Categorised\_in (<FcID>, <DishID>)

CSDish (<DishID>, Price)

Orders (<PgrID>, <DishID>, OrderPrice, OrderQty, OrderDateTime, DeliverTo, DelDateTime)

# Data Dictionary

**Passenger** Relation: Stores the information of the passengers travelling on the cruise ship

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute**  **Name** | **Description** | **Data Type and Length** | **Constraint** | **Null Value** |
| PgrID | Unique ID of each passenger of MokeCruise | CHAR(4) | Primary Key | Not Null |
| PgrName | Name of passenger | VARCHAR(150) |  | Not Null |
| PgrGender | Gender of the passenger | CHAR(1) | Values : M for Male, F for Female | Not Null |
| PgrEmail | Email address of the passenger | VARCHAR(150) |  | Not Null |
| PgrDOB | Date of birth of the passenger | DATE | DOB not later than current date | Not Null |
| CabinNo | Cabin room number of the passengers. | CHAR(4) | A cabin may have up to 4 passengers. | Not Null |

**ContactNumber** Relation: Stores contact number of each passenger

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute**  **Name** | **Description** | **Data Type and Length** | **Constraint** | **Null Value** |
| PgrID | Identifies the passenger with the mobile number | CHAR(4) | Primary Key  Foreign Key => Passenger (PgrID) | Not Null |
| PgrContactNo | Mobile number of the passenger | CHAR(8) | Primary Key | Not Null |

**Reservation** Relation: Stores booking placed by passenger for dining in an eatery

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute**  **Name** | **Description** | **Data Type and Length** | **Constraint** | **Null Value** |
| ReservID | Unique identifier for each reservation made | CHAR(4) | Primary Key | Not Null |
| ReservDateTime | Date and time when the booking is made for the eatery is made | DATETIME | Maximum 1 day before RequiredDateTime | Not Null |
| RequiredDateTime | Date and time when passengers will dine at the eatery | DATETIME |  | Not Null |
| NoOfPax | Number of passengers who will be dining at the eatery. | TINYINT |  | Not Null |
| ReservStatus | Status of the Reservation | VARCHAR(150) | Values : Booked, Confirmed, Wait List  Defaulted to Booked | Not Null |
| PgrID | Identify the Passenger which made the reservation | CHAR(4) | Foreign Key => Passenger(PgrID) | Not Null |
| EatyID | Identifiies the eatery booked for dining | CHAR(4) | Foreign Key =>  Eatery(EatyID) | Not Null |

**Booking** Relation: Stores booking placed by passenger for a session of an event

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute**  **Name** | **Description** | **Data Type and Length** | **Constraint** | **Null Value** |
| BookingID | The unique identifier of the booking made in the system | CHAR(4) | Primary Key | Not Null |
| NoOfAdultTicket | Number of adult tickets required for participation in the event. | TINYINT |  | Not Null |
| AdultSalesPrice | Actual price paid for adult participating in event | SMALLMONEY |  | Null |
| NoOfChildTicket | Number of child tickets required for participation in the event | TINYINT |  | Not Null |
| ChildSalesPrice | Actual price paid for a child participating in the event | SMALLMONEY |  | Null |
| BookDateTime | Date and time when the booking is made for the event | DATETIME | Maximum 1 day before event date | Not Null |
| BookStatus | Status of the booking | VARCHAR(150) | Values : Booked, Confirmed, Wait List  Defaulted to Booked | Not Null |
| EventID | Identifies the event booked | CHAR(4) | Foreign Key => Event(EventID) | Not Null |
| SessionNo | Identifies the session for the event booked | TINYINT | Foreign Key =>  EventSession(SessionNo) | Not Null |
| PgrID | Identifies the passenger who made the booking | CHAR(4) | Foreign key =>  Passenger(PgrID) | Not Null |

**Event** Relation: Stores the booking of the events that are available for passsengers to enjoy

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute**  **Name** | **Description** | **Data Type and Length** | **Constraint** | **Null Value** |
| EventID | Unique identifier of the event | CHAR(4) | Primary Key | Not Null |
| EventName | Name of event | VARCHAR(150) |  | Not Null |
| EventDescr | Description of the event | VARCHAR(150) |  | Not Null |
| EventCapacity | Capacity of the event | TINYINT |  | Not Null |
| EventDuration | Duration of the event in hours | TINYINT |  | Not Null |
| EventLoc | Location of the event | VARCHAR(150) |  | Not Null |
| MinAge | Minimum age of the participant of the event | TINYINT |  | Null |
| MaxAge | Maximum age of the participant of the event | TINYINT |  | Null |
| AdultPrice | Price a adult has to pay to participate in paid events | SMALLMONEY |  | Null |
| ChildPrice | Price a child has to pay to participate in paid events | SMALLMONEY |  | Null |
| ETID | Identifies the event type | CHAR(4) | Foreign Key =>  EventType (ETID) | Not Null |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute**  **Name** | **Description** | **Data Type and Length** | **Constraint** | **Null Value** |
| EventID | Identifies the event | CHAR(4) | Primary Key  Foreign Key =>  Event(EventID) | Not Null |
| SessionNo | A serial number that identifies a session/run of the event | TINYINT | Primary Key | Not Null |
| EventDateTime | Date and Time the event will start | DATETIME |  | Not Null |

**EventSession** Relation: Stores the session number and the date and time of the event

**EventType** Relation: Stores the type of event available

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute**  **Name** | **Description** | **Data Type and Length** | **Constraint** | **Null Value** |
| ETID | Unique identifier of the event type | CHAR(4) | Primary Key | Not Null |
| ETName | Name of the type of event | VARCHAR(150) |  | Not Null |

**Eatery** Relation: Stores the information of the restaurant and dining hall

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute**  **Name** | **Description** | **Data Type and Length** | **Constraint** | **Null Value** |
| EatyID | Unique identifier of the restaurant or dining hall | CHAR(4) | Primary Key | Not Null |
| EatyName | Name of restaurant or dining hall | VARCHAR(150) |  | Not Null |
| EatyClHr | Closing hour of the restaurant or dining hall | TIME | After the opening hour | Not Null |
| EatyOpHr | Opening hour of the restaurant or dining hall | TIME |  | Not Null |
| EatyLoc | Location of the restaurant or dining hall | VARCHAR(150) |  | Null |
| EatyCapacity | Capacity of the restaurant or dining hall | TINYINT |  | Null |

**Dish** Relation: Stores the information of the dishes available on the cruise shihp

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute**  **Name** | **Description** | **Data Type and Length** | **Constraint** | **Null Value** |
| DishID | Unique identifier of the dish | CHAR(4) | Primary Key | Not Null |
| DishName | Name of the dish | VARCHAR(150) |  | Not Null |
| DishDescr | Description of the dish | VARCHAR(150) |  | Not Null |
| EatyID | Unique identifier of the restaurant serving the dish | CHAR(4) | Foreign Key =>  Eatery (EatyID) | Not Null |
| CuisineID | Unique identifier of the cuisine of the dish | CHAR(4) | Foreign Key => Cuisine (CuisineID) | Not Null |

**Ingredient** Relation: Stores the information of the ingredients

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute**  **Name** | **Description** | **Data Type and Length** | **Constraint** | **Null Value** |
| IngredID | Unique identifier of ingredients used | CHAR(4) | Primary Key | Not Null |
| IngredName | Name of ingredient | VARCHAR(150) |  | Not Null |

**Contain** Relation: Stores the list of ingredients used in the dish

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute**  **Name** | **Description** | **Data Type and Length** | **Constraint** | **Null Value** |
| DishID | Unique identifier of dish | CHAR(4) | Primary Key  Foreign Key =>  Dish (DishID) | Not Null |
| IngredID | Unique identifier of ingredients used in the dish | CHAR(4) | Primary Key  Foreign Key =>  Ingredient (IngredID) | Not Null |

**Cuisine** Relation: Stores the style of cooking

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute**  **Name** | **Description** | **Data Type and Length** | **Constraint** | **Null Value** |
| CuisineID | Unique identifier of cuisine | CHAR(4) | Primary Key | Not Null |
| CuisineName | Name of cuisine | VARCHAR(150) |  | Not Null |

**FoodCategory** Relation: Stores the type of dish

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute**  **Name** | **Description** | **Data Type and Length** | **Constraint** | **Null Value** |
| FcID | Unique identifier of the food category | CHAR(4) | Primary Key | Not Null |
| FcName | Name of the food category | VARCHAR(150) |  | Not Null |
| FcDescr | Description of the food category of the dish | VARCHAR(150) |  | Not Null |

**Categorised\_in** Relation: Stores the food category of the dish

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute**  **Name** | **Description** | **Data Type and Length** | **Constraint** | **Null Value** |
| FcID | Unique identifier of the food category | CHAR(4) | Primary Key  Foreign Key =>  FoodCategory(FcID) | Not Null |
| DishID | Unique identifier of dish | CHAR(4) | Primary Key  Foreign Key =>  Dish(DishID) | Not Null |

**CSDish** Relation: Stores the dish information that is available only for cabin service

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute**  **Name** | **Description** | **Data Type and Length** | **Constraint** | **Null Value** |
| DishID | Unique identifier of the dish | CHAR(4) | Primary Key  Foreign Key =>  Dish(DishID) | Not Null |
| Price | Price of the cabin service dish | SMALLMONEY |  | Not Null |

**Orders** Relation:Stores the information of cabin service ordered by passengers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute**  **Name** | **Description** | **Data Type and Length** | **Constraint** | **Null Value** |
| PgrID | Unique identifier of passenger who ordered the dish | CHAR(4) | Primary Key  Foreign Key => Passenger(PgrID) | Not Null |
| DishID | Unique identifier of dish ordered | CHAR(4) | Primary Key  Foreign Key => Dish(DishID) | Not Null |
| OrderPrice | Actual price paid for cabin service dish ordered | SMALLMONEY |  | Not Null |
| OrderQty | Quantity of cabin service dishes ordered | TINYINT |  | Not Null |
| OrderDateTime | Date and time cabin service is ordered | DATETIME | Primary Key  Not later than current date  Default to Today | Not Null |
| DeliverTo | CabinNo cabin service is delivered to | CHAR(4) | CabinNo | Not Null |
| DelDateTime | Date and time the cabin service is delivered to the cabin | DATETIME |  | Null |

# SQL Commands

## SQL Commands for Tables Creation

/\*\*\* Delete database if it exists \*\*\*/

USE master

IF EXISTS(select \* from sys.databases where name='EnDine')

DROP DATABASE EnDine

GO

Create Database EnDine

GO

use EnDine

GO

/\*\*\* Delete tables (if they exist) before creating \*\*\*/

/\* Table: Orders \*/

if exists (select \* from sysobjects

where id = object\_id('Orders') and sysstat & 0xf = 3)

drop table Orders

GO

/\* Table: CSDish \*/

if exists (select \* from sysobjects

where id = object\_id('CSDish') and sysstat & 0xf = 3)

drop table CSDish

GO

/\* Table: Categorised\_in \*/

if exists (select \* from sysobjects

where id = object\_id('Categorised\_in') and sysstat & 0xf = 3)

drop table Categorised\_in

GO

/\* Table: FoodCategory \*/

if exists (select \* from sysobjects

where id = object\_id('FoodCategory') and sysstat & 0xf = 3)

drop table FoodCategory

GO

/\* Table: Contain \*/

if exists (select \* from sysobjects

where id = object\_id('Contain') and sysstat & 0xf = 3)

drop table Contain

GO

/\* Table: Ingredient \*/

if exists (select \* from sysobjects

where id = object\_id('Ingredient') and sysstat & 0xf = 3)

drop table Ingredient

GO

/\* Table: Dish \*/

if exists (select \* from sysobjects

where id = object\_id('Dish') and sysstat & 0xf = 3)

drop table Dish

GO

/\* Table: Cuisine \*/

if exists (select \* from sysobjects

where id = object\_id('Cuisine') and sysstat & 0xf = 3)

drop table Cuisine

GO

/\* Table: Reservation \*/

if exists (select \* from sysobjects

where id = object\_id('Reservation') and sysstat & 0xf = 3)

drop table Reservation

GO

/\* Table: Eatery \*/

if exists (select \* from sysobjects

where id = object\_id('Eatery') and sysstat & 0xf = 3)

drop table Eatery

GO

/\* Table: Booking \*/

if exists (select \* from sysobjects

where id = object\_id('Booking') and sysstat & 0xf = 3)

drop table Booking

GO

/\* Table: ContactNumber \*/

if exists (select \* from sysobjects

where id = object\_id('ContactNumber') and sysstat & 0xf = 3)

drop table ContactNumber

GO

/\* Table: Passenger \*/

if exists (select \* from sysobjects

where id = object\_id('Passenger') and sysstat & 0xf = 3)

drop table Passenger

GO

/\* Table: EventSession \*/

if exists (select \* from sysobjects

where id = object\_id('EventSession') and sysstat & 0xf = 3)

drop table EventSession

GO

/\* Table: Event \*/

if exists (select \* from sysobjects

where id = object\_id('Event') and sysstat & 0xf = 3)

drop table Event

GO

/\* Table: EventType \*/

if exists (select \* from sysobjects

where id = object\_id('EventType') and sysstat & 0xf = 3)

drop table EventType

GO

/\*\*\* Create tables \*\*\*/

/\* Table: EventType \*/

CREATE TABLE EventType

(

ETID CHAR(4) NOT NULL,

ETName VARCHAR(150) NOT NULL,

CONSTRAINT PK\_EventType PRIMARY KEY NONCLUSTERED (ETID)

)

GO

/\* Table: Event \*/

CREATE TABLE Event

(

EventID CHAR(4) NOT NULL,

EventName VARCHAR(150) NOT NULL,

EventDescr VARCHAR(150) NOT NULL,

EventCapacity TINYINT NOT NULL,

EventDuration TINYINT NOT NULL,

EventLoc VARCHAR(150) NOT NULL,

MinAge TINYINT NULL,

MaxAge TINYINT NULL,

AdultPrice SMALLMONEY NULL,

ChildPrice SMALLMONEY NULL,

ETID CHAR(4) NOT NULL,

CONSTRAINT PK\_Event PRIMARY KEY NONCLUSTERED (EventID),

CONSTRAINT FK\_Event\_ETID FOREIGN KEY (ETID) REFERENCES EventType(ETID)

)

GO

/\* Table: EventSession \*/

CREATE TABLE EventSession

(

EventID CHAR(4) NOT NULL,

SessionNo TINYINT NOT NULL,

EventDateTime DATETIME NOT NULL,

CONSTRAINT PK\_EventSession PRIMARY KEY NONCLUSTERED (EventID, SessionNo),

CONSTRAINT FK\_EventSession\_EventID FOREIGN KEY (EventID) REFERENCES Event(EventID)

)

GO

/\* Table: Passenger \*/

CREATE TABLE Passenger

(

PgrID CHAR(4) NOT NULL,

PgrName VARCHAR(150) NOT NULL,

PgrGender CHAR(1) NOT NULL CHECK(PgrGender IN ('M','F')),

PgrEmail VARCHAR(150) NOT NULL,

PgrDOB DATE NOT NULL CHECK(PgrDOB < GETDATE()),

CabinNo CHAR(4) NOT NULL,

CONSTRAINT PK\_Passenger PRIMARY KEY NONCLUSTERED (PgrID),

)

GO

/\* Table: ContactNumber \*/

CREATE TABLE ContactNumber

(

PgrID CHAR(4) NOT NULL,

PgrContactNo CHAR(8) NOT NULL,

CONSTRAINT PK\_ContactNumber PRIMARY KEY NONCLUSTERED (PgrID, PgrContactNo),

CONSTRAINT FK\_ContactNumber\_PgrID FOREIGN KEY (PgrID) REFERENCES Passenger(PgrID)

)

GO

/\* Table: Booking \*/

CREATE TABLE Booking

(

BookingID CHAR(4) NOT NULL,

NoOfAdultTicket TINYINT NOT NULL,

AdultSalesPrice SMALLMONEY Null,

NoOfChildTicket TINYINT NOT NULL,

ChildSalesPrice SMALLMONEY NULL,

BookDateTime DATETIME NOT NULL DEFAULT(GETDATE()),

BookStatus VARCHAR(150) NOT NULL DEFAULT('Booked') CHECK(BookStatus IN ('Booked', 'Confirmed', 'Wait List')),

EventID CHAR(4) NOT NULL,

SessionNo TINYINT NOT NULL,

PgrID CHAR(4) NOT NULL,

CONSTRAINT PK\_Booking PRIMARY KEY NONCLUSTERED (BookingID),

CONSTRAINT FK\_Booking\_SessionNo FOREIGN KEY (EventID, SessionNo) REFERENCES EventSession(EventID, SessionNo),

CONSTRAINT FK\_Booking\_PgrID FOREIGN KEY (PgrID) REFERENCES Passenger(PgrID),

--CONSTRAINT CHK\_Booking CHECK (DATEPART(Day,(SELECT EventDateTime FROM EventSession WHERE SessionNo = Booking.SessionNo AND EventID = Booking.EventID),BookDateTime) <= 1)

)

GO

/\* Table: Eatery \*/

CREATE TABLE Eatery

(

EatyID CHAR(4) NOT NULL,

EatyName VARCHAR(150) NOT NULL,

EatyClHr TIME NOT NULL,

EatyOpHr TIME NOT NULL,

EatyLoc VARCHAR(150) NULL,

EatyCapacity TINYINT NULL,

CONSTRAINT PK\_Eatery PRIMARY KEY NONCLUSTERED (EatyID),

CONSTRAINT CHK\_Eatery CHECK (EatyClHr > EatyOpHr)

)

GO

/\* Table: Reservation \*/

CREATE TABLE Reservation

(

ReservID CHAR(4) NOT NULL,

ReservDateTime DATETIME NOT NULL,

RequiredDateTime DATETIME NOT NULL,

NoOfPax TINYINT NOT NULL,

ReservStatus VARCHAR(150) NOT NULL DEFAULT('Booked') CHECK(ReservStatus IN ('Booked', 'Confirmed', 'Wait List')),

PgrID CHAR(4) NOT NULL,

EatyID CHAR(4) NOT NULL,

CONSTRAINT PK\_Reservation PRIMARY KEY NONCLUSTERED (ReservID),

CONSTRAINT FK\_Reservation\_PgrID FOREIGN KEY (PgrID) REFERENCES Passenger(PgrID),

CONSTRAINT FK\_Reservation\_EatyID FOREIGN KEY (EatyID) REFERENCES Eatery(EatyID),

--CONSTRAINT CHK\_Reservation CHECK (DATEDIFF(DAY,ReservDateTime, RequiredDateTime) <= 1)

)

GO

/\* Table: Cuisine \*/

CREATE TABLE Cuisine

(

CuisineID CHAR(4) NOT NULL,

CuisineName VARCHAR(150) NOT NULL,

CONSTRAINT PK\_Cuisine PRIMARY KEY NONCLUSTERED (CuisineID)

)

GO

/\* Table: Dish \*/

CREATE TABLE Dish

(

DishID CHAR(4) NOT NULL,

DishName VARCHAR(150) NOT NULL,

DishDescr VARCHAR(150) NOT NULL,

EatyID CHAR(4) NOT NULL,

CuisineID CHAR(4) NOT NULL,

CONSTRAINT PK\_Dish PRIMARY KEY NONCLUSTERED (DishID),

CONSTRAINT FK\_Dish\_EatyID FOREIGN KEY (EatyID) REFERENCES Eatery(EatyID)

)

GO

/\* Table: Ingredient \*/

CREATE TABLE Ingredient

(

IngredID CHAR(4) NOT NULL,

IngredName VARCHAR(150) NOT NULL,

CONSTRAINT PK\_Ingredient PRIMARY KEY NONCLUSTERED (IngredID)

)

GO

/\* Table: Contain \*/

CREATE TABLE Contain

(

DishID CHAR(4) NOT NULL,

IngredID CHAR(4) NOT NULL,

CONSTRAINT PK\_Contain PRIMARY KEY NONCLUSTERED (DishID, IngredID),

CONSTRAINT FK\_Contain\_DishID FOREIGN KEY (DishID) REFERENCES Dish(DishID),

CONSTRAINT FK\_Contain\_IngredID FOREIGN KEY (IngredID) REFERENCES Ingredient(IngredID)

)

GO

/\* Table: FoodCategory \*/

CREATE TABLE FoodCategory

(

FcID CHAR(4) Primary Key NONCLUSTERED,

FcName VARCHAR(150) NOT NULL,

FcDescr VARCHAR(150) NOT NULL,

)

GO

/\* Table: Categorised\_in \*/

CREATE TABLE Categorised\_in

(

FcID CHAR(4) NOT NULL,

DishID CHAR(4) NOT NULL,

CONSTRAINT PK\_FoodCategory PRIMARY KEY NONCLUSTERED (FcID, DishID),

CONSTRAINT FK\_FoodCategory\_FcID FOREIGN KEY (FcID) REFERENCES FoodCategory(FcID),

CONSTRAINT FK\_FoodCategory\_DishID FOREIGN KEY (DishID) REFERENCES Dish(DishID)

)

GO

/\* Table: CSDish \*/

CREATE TABLE CSDish

(

DishID CHAR(4) NOT NULL,

Price SMALLMONEY NOT NULL,

CONSTRAINT PK\_CSDish PRIMARY KEY NONCLUSTERED (DishID),

CONSTRAINT FK\_CSDish\_DishID FOREIGN KEY (DishID) REFERENCES Dish(DishID)

)

GO

/\* Table: Orders \*/

CREATE TABLE Orders

(

PgrID CHAR(4) NOT NULL,

DishID CHAR(4) NOT NULL,

OrderPrice SMALLMONEY NOT NULL,

OrderQty TINYINT Not NULL,

OrderDateTime DATETIME NOT NULL DEFAULT(GETDATE()),

DeliverTo CHAR(4) NOT NULL,

DelDateTime DATETIME NULL,

CONSTRAINT PK\_Orders PRIMARY KEY NONCLUSTERED (PgrID, DishID, OrderDateTime),

CONSTRAINT FK\_Orders\_PgrID FOREIGN KEY (PgrID) REFERENCES Passenger(PgrID),

CONSTRAINT FK\_Orders\_DishID FOREIGN KEY (DishID) REFERENCES Dish(DishID),

--CONSTRAINT CHK\_Orders CHECK (DeliverTo IN Passenger(CabinNo)),

CONSTRAINT CHK\_Orders CHECK (OrderDateTime <= GETDATE())

)

GO

## SQL Commands for Insertion of Data

/\* Insert rows \*/

INSERT INTO EventType(ETID,ETName) VALUES ('A001', 'Performances'),

('A002', 'Films and Movies'),

('A003', 'Fitness'),

('A004', 'Fun and Games'),

('A005', 'Sports and Adventure'),

('A006', 'Cultural'),

('A007', 'Rest and Relaxation'),

('A008', 'Conferences'),

('A009', 'Dancing'),

('A010', 'Educational')

INSERT INTO Event(EventID, EventName, EventDescr, EventCapacity, EventDuration, EventLoc, MinAge, MaxAge, AdultPrice, ChildPrice, ETID)

VALUES ('E001', 'Standup Comedy', 'listen to a comedic experience in our day to day lives', 50, 0.5, 'The Enigma Concert Hall', NULL, NULL, NULL, NULL, 'A001'),

('E002', 'Disney on ice', 'Travel on an adventure with the Disney princesses', 100, 2, 'Sensational Ice World', NULL, NULL, 10, 5, 'A001'),

('E003', 'Sing2', 'Watch an all-star cast of animal performers prepare to launch a dazzling stage extravaganza in the glittering entertainment capital of the world', 50, 1.84, 'Theatre 1', NULL, NULL, NULL, NULL, 'A002'),

('E004', 'The 355', 'A fast-paced globe-trotting espionage thriller that brings together a dream team of formidable female stars.', 50, 2.05, 'Theatre 2', 13, NULL, NULL, NULL, 'A002'),

('E005', 'The King’s Man', 'In the early years of the 20th century, the Kingsman agency is formed to stand against a cabal plotting a war to wipe out millions.', 50, 2.20, 'Theatre 3', 16, NULL, NULL, NULL, 'A002'),

('E006', 'Spider-Man: No Way Home', 'Our friendly neighborhood hero is unmasked and no longer able to separate his normal life from the high-stakes of being a Super Hero', 50, 2.50, 'Theatre 4', NULL, NULL, NULL, NULL, 'A002'),

('E007', 'Gym', 'Access top tiered Gym equipment to maintain your fitness', 80, 2, 'Gym', 18, NULL, NULL, NULL, 'A003'),

('E008', 'Pilates', 'Develop the body through muscular effort that stems from the core', 30, 2, 'Gym Room 1', 5, 80, 5, NULL, 'A003'),

('E009', 'Scavenger Hunt', 'Bond with your family and friends as you work together to find items to solve mysteries', 60, 1, 'Ball Room 1', NULL, NULL, NULL, NULL, 'A004'),

('E010', 'Laser Tag', 'Bond with your family and friends as you aim to top the leaderboard', 20, 1, 'LaseRise', 6, 80, NULL, NULL, 'A004'),

('E011', 'Rock Climbing', 'Scale the wall (3 levels high), challenge your friends and see who reaches first', 150, 2, 'Rock Wall', 5, 60, NULL, NULL, 'A005'),

('E012', 'Adventure Cove', 'Enjoy the iconic water theme park', 80, 4, 'Adventure Cove', NULL, NULL, 10, 5, 'A005'),

('E013', 'Ice Skating', 'Skate around the ring with your friends and family', 25, 1, 'Sensational Ice World', 5, NULL, NULL, NULL, 'A005'),

('E014', 'Roller Blading', 'Roll your way around the pier with your friends and family', 60, 1, 'Pier', 3, NULL, NULL, NULL, 'A005'),

('E015', 'WaveBoarding', 'Surf like you have never done so before, learn from the experts tips and tricks in wave boarding', 7, 0.5, 'FlowRider', 7, 60, NULL, NULL, 'A005'),

('E016', 'Mini Golf', 'Test your golfing skills on our minature golf course', 20, 1, 'Take Your Swing', NULL, NULL, NULL, NULL, 'A005'),

('E017', 'Wine History', 'Learn more about the history of wine and taste the different variety of wine', 25, 2, 'Winery', 18, NULL, 20, NULL, 'A006'),

('E018', 'Tea Observation Ceromony', 'Observe how the chinese takes care of their tea pots', 20, 1, 'Museum', NULL, NULL, NULL, NULL, 'A006'),

('E019', 'Spa Session', 'Enjoy a relaxing spa with hot stones', 20, 1, 'Spa', NULL, NULL, NULL, NULL, 'A007'),

('E020', 'Chiropractic', 'Relax your muscles and relieve all stresses on your body', 20, 1, 'Spa', NULL, NULL, NULL, NULL, 'A007'),

('E021', 'Technology Conference', 'Learn more about the advancement of technology in the industry', 100, 1, 'Ball Room 2', NULL, NULL, NULL, NULL, 'A008'),

('E022', 'Cyber Security Conference', 'Hear sessions about the latest cyber threats', 100, 1, 'Ball Room 2', NULL, NULL, NULL, NULL, 'A008'),

('E023', 'Anime Conference', 'Learn and find new friends who share a common interest in anime', 100, 1, 'Ball Room 2', NULL, NULL, NULL, NULL, 'A008'),

('E024', 'Mordern Dance Class', 'Learn the dance style with mixed influences of other dance styles', 50, 2, 'Ball Room 1', 6, 50, NULL, NULL, 'A009'),

('E025', 'Tango Dance', 'Experience the partner dance and social dance that originated in the 1880s', 50, 1, 'Ball Room 2', NULL, NULL, NULL, NULL, 'A009'),

('E026', 'Cha Cha Cha Dance Class', 'Learn the dance of Cuban origin', 50, 1, 'Ball Room 1', NULL, NULL, NULL, NULL, 'A009'),

('E027', 'Jazz Dance Class', 'Learn social dance style that emerged at the turn of the 20th century', 30, 2, 'Ball Room 2', NULL, NULL, NULL, NULL, 'A009'),

('E028', 'Cooking Class', 'Cook your way to the top of the class with our Michelin star chefs', 30, 1, 'Learning Kitchen', NULL, NULL, NULL, NULL, 'A010'),

('E029', 'Theatric class', 'Express yourself through the art form of acting', 30, 1, 'The Enigma Concert Hall', NULL, NULL, NULL, NULL, 'A010'),

('E030', 'Art class', 'Hop into your creative mind and explore as you unleash your creative energy', 30, 1, 'Pier', NULL, NULL, NULL, NULL, 'A010'),

('E031', 'Science class', 'Experience Fossil digs. Junior chemistry. Space Mud making', 20, 1, 'Experential Learning Lab', NULL, NULL, 8, NULL, 'A010')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E001', 1, '2022-01-15 10:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E001', 2, '2022-01-16 10:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E002', 1, '2022-01-15 13:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E002', 2, '2022-01-16 13:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E003', 1, '2022-01-15 11:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E003', 2, '2022-01-16 11:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E003', 3, '2022-01-17 11:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E004', 1, '2022-01-15 09:30:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E004', 2, '2022-01-16 14:30:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E004', 3, '2022-01-17 17:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E005', 1, '2022-01-15 12:30:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E005', 2, '2022-01-16 12:30:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E005', 3, '2022-01-17 12:30:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E006', 1, '2022-01-15 16:30:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E006', 2, '2022-01-16 16:30:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E007', 1, '2022-01-15 08:30:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E007', 2, '2022-01-16 08:30:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E008', 1, '2022-01-15 16:30:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E008', 2, '2022-01-16 16:30:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E009', 1, '2022-01-15 08:30:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E009', 2, '2022-01-16 08:30:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E009', 3, '2022-01-17 08:30:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E010', 1, '2022-01-15 14:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E010', 2, '2022-01-16 14:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E010', 3, '2022-01-17 14:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E011', 1, '2022-01-15 11:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E011', 2, '2022-01-16 11:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E012', 1, '2022-01-15 08:30:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E012', 2, '2022-01-16 08:30:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E013', 1, '2022-01-15 08:30:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E013', 2, '2022-01-16 08:30:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E014', 1, '2022-01-15 20:30:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E014', 2, '2022-01-16 20:30:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E015', 1, '2022-01-15 14:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E015', 2, '2022-01-16 14:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E015', 3, '2022-01-17 09:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E015', 4, '2022-01-17 14:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E016', 1, '2022-01-15 14:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E016', 2, '2022-01-16 14:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E017', 1, '2022-01-15 17:30:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E018', 1, '2022-01-15 14:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E018', 2, '2022-01-16 14:30:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E019', 1, '2022-01-15 12:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E019', 2, '2022-01-16 12:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E020', 1, '2022-01-15 16:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E020', 2, '2022-01-16 16:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E021', 1, '2022-01-15 08:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E021', 2, '2022-01-16 08:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E022', 1, '2022-01-15 10:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E022', 2, '2022-01-16 10:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E023', 1, '2022-01-15 13:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E023', 2, '2022-01-16 13:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E024', 1, '2022-01-15 08:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E024', 2, '2022-01-17 08:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E025', 1, '2022-01-15 17:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E025', 2, '2022-01-17 17:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E026', 1, '2022-01-15 15:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E026', 2, '2022-01-16 15:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E026', 3, '2022-01-17 15:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E027', 1, '2022-01-15 20:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E027', 2, '2022-01-16 20:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E028', 1, '2022-01-15 09:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E028', 2, '2022-01-17 09:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E029', 1, '2022-01-15 09:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E029', 2, '2022-01-16 09:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E030', 1, '2022-01-15 16:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E030', 2, '2022-01-16 17:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E030', 3, '2022-01-17 12:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E031', 1, '2022-01-15 15:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E031', 2, '2022-01-16 09:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E031', 3, '2022-01-16 15:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E031', 4, '2022-01-17 09:00:00')

INSERT INTO EventSession(EventID, SessionNo, EventDateTime) VALUES ('E031', 5, '2022-01-17 15:00:00')

INSERT INTO Passenger(PgrID, PgrName, PgrGender, PgrEmail, PgrDOB, CabinNo)

VALUES ('0001', 'Anya Tay', 'F', 'anya@hotmail.com', '2000-02-29', 'C001'),

('0002', 'Reyna Tay', 'F', 'reyna@hotmail.com', '2002-07-30', 'C001'),

('0003', 'Leyla Vang', 'F', 'leya@gmail.com', '1970-06-03', 'C002'),

('0004', 'Belinda Taylor', 'F', 'bt@outlook.com', '1985-04-16', 'C003'),

('0005', 'Heidi Navarro', 'F', 'heidnev@ymail.com', '1995-01-23', 'C004'),

('0006', 'Amy Santiago', 'F', 'amy@outlook.com', '1984-12-31', 'C004'),

('0007', 'Elena Lyons', 'F', 'l\_elena@outlook.com', '1984-12-25', 'C004'),

('0008', 'Ashleigh Mosley', 'F', 'ashleigh@ymail.com', '1966-09-07', 'C005'),

('0009', 'Camilla Cabello', 'F', 'camilla@gmail.com', '2011-06-16', 'C005'),

('0010', 'Gloria Wilcox', 'F', 'gloriaw@hotmail.com', '1973-11-21', 'C006'),

('0011', 'Chelsea Holloway', 'F', 'chel@ymail.com', '1984-01-24', 'C007'),

('0012', 'Avery Beck', 'F', 'avebeck@hotmail.com', '2004-03-18', 'C008'),

('0013', 'Lilia Perez', 'F', 'perez\_lilia@outlook.com', '2005-06-02', 'C008'),

('0014', 'Amber Day', 'F', 'ombreamber@gmail.com', '1966-10-25', 'C009'),

('0015', 'Mia Doyle', 'F', 'doyle\_mia@hotmail.com', '2013-10-09', 'C009'),

('0016', 'Kendall Abbott', 'F', 'abbotkendall@yahoo.com', '1975-06-30', 'C009'),

('0017', 'Evangeline Vaughn', 'F', 'vaughneva@yahoo.com', '2013-07-27', 'C010'),

('0018', 'Shyanne Dennis', 'F', 'shyanne@gmail.com', '1968-07-15', 'C010'),

('0019', 'Sierra Heath', 'F', 'sierra@yahoo.com', '1974-12-21', 'C011'),

('0020', 'Alia Mullins', 'F', 'aliam@outlook.com', '1964-08-13', 'C012'),

('0021', 'Joselyn Grimes', 'F', 'joselyng@ymail.com', '1965-04-01', 'C012'),

('0022', 'Hazel Cobb', 'F', 'hazel@outlook.com', '2008-04-06', 'C012'),

('0023', 'Alyson Beasley', 'F', 'alyson@hotmail.com', '1990-06-25', 'C013'),

('0024', 'Miah Lim', 'F','miah@yahoo.com', '1965-02-25', 'C013'),

('0025', 'Reyna Chua', 'F', 'reyc@gmail.com', '1971-10-14', 'C014'),

('0026', 'Lim Meng Sian Keanu', 'M', 'keanu@hotmail.com', '2011-09-17', 'C014'),

('0027', 'Colin Foo Chi Yang', 'M', 'cfcy@gmail.com', '1997-07-03', 'C015'),

('0028', 'Ng E Kai', 'M', 'ekagi@hotmail.com', '2016-07-29', 'C015'),

('0029', 'Kyler Lee', 'M', 'kyler@outlook.com', '1998-07-07', 'C016'),

('0030', 'Arulraj Jonse', 'M', 'jonse@ymail.com', '2008-04-17', 'C016'),

('0031', 'Chong Wei Qin', 'M', 'sirarthur@gmail.com', '2019-11-20', 'C016'),

('0032', 'Soh Yi Wei', 'M', 'sohyw@gmail.com', '1997-09-26', 'C017'),

('0033', 'Qin Tian', 'M', 'clouduqin@gmail.com', '2008-03-24', 'C017'),

('0034', 'Qin Guan', 'M', 'iguanaqin@hotmail.com', '1970-01-18', 'C018'),

('0035', 'Goh Seng Ern Brendan', 'M', 'saggysocks@gmail.com', '2001-03-31', 'C018'),

('0036', 'Cyrus Tan Rui Xuan', 'M', 'cyrus@hotmail.com', '1971-04-21', 'C019'),

('0037', 'Tan Kai Zhe', 'M', 'tkz@icloud.com', '2013-11-13', 'C019'),

('0038', 'Enrico Tionardi Owen', 'M', 'enrico@gmail.com', '1979-11-27', 'C020'),

('0039', 'Jeow Teng Xiang Cody', 'M', 'cody@gmail.com', '1977-09-17', 'C020'),

('0040', 'Tay Gao Jun', 'M', 'cattolation@icloud.com', '1977-09-25', 'C021'),

('0041', 'Benedict Woo Jun Kai', 'M', 'ben10@hotmail.com', '1986-06-27', 'C022'),

('0042', 'Clement Neo Jun Kai', 'M', 'hamster@outlook.com', '1983-12-31', 'C023'),

('0043', 'Yong Zi Ren', 'M', 'yzr@outlook.com', '1972-08-29', 'C024'),

('0044', 'Suen Jeren', 'M', 'jeren@hotmail.com', '1965-08-09', 'C024'),

('0045', 'Neo Jia Chen', 'M', 'jiachen@gmail.com', '1968-09-25', 'C025'),

('0046', 'Lim Long Teck', 'M', 'llt@yahoo.com', '1978-08-06', 'C026'),

('0047', 'Lor Jing Xiang Eugene', 'M', 'eugene@ymail.com', '1979-03-31', 'C026'),

('0048', 'Goh Tian Le Mathew', 'M', 'matthew@hotmail.com', '1990-09-26', 'C027'),

('0049', 'Deshawn Clayton', 'M', 'deshawn@icloud.com', '1975-04-11', 'C028'),

('0050', 'keene Ng', 'M', 'keene@outlook.com', '1975-11-10', 'C029'),

('0051', 'Tan Yee Ming', 'F', 'yeetming@gmail.com', '2012-05-25', 'C030'),

('0052', 'Tan Yun E', 'F', 'cloud@gmail.com', '1966-02-06', 'C008'),

('0053', 'Kayley Perez', 'F', 'kayleyp@gmail.com', '1965-10-04', 'C030'),

('0054', 'Josephine Crawford', 'F', 'jcrawford@outlook.com', '1990-05-27', 'C031'),

('0055', 'Wendy Hicks', 'F', 'wendyh@ymail.com', '1984-08-17', 'C032'),

('0056', 'Piper Holden', 'F', 'piperh@yahoo.com', '1976-10-01', 'C033'),

('0057', 'Payton Camacho', 'F', 'payton@ymail.com', '1973-05-10', 'C034'),

('0058', 'Scarlett Reilly', 'F', 'reilly\_scarlett@gmail.com', '1970-05-13', 'C035'),

('0059', 'Tan Pei Yee', 'F', 'peiyee@outlook.com', '1964-05-27', 'C036'),

('0060', 'Mok Qing Ling', 'F', 'qlmok@outlook.com', '1988-12-06', 'C037'),

('0061', 'Choo Cheng How', 'M', 'cch5@hotmail.com', '1978-07-30', 'C038'),

('0062', 'Leroi June Yan Hao', 'M', 'leroijune@ymail.com', '2001-03-12', 'C039'),

('0063', 'Lee Yu Yee Dominic', 'M', 'dom@outlook.com', '1975-07-26', 'C004'),

('0064', 'Alfred Leandro Liang', 'M', 'alfred@gmail.com', '1965-06-10', 'C040'),

('0065', 'Kyler Lee', 'M', 'kyler@hotmail.com', '2010-06-10', 'C040'),

('0066', 'Lee Cheng Jhang', 'M', 'cj96@outlook.com', '1996-07-01', 'C041'),

('0067', 'Han Xihe', 'M', 'hxh@ymail.com', '1984-03-19', 'C041'),

('0068', 'Keefe Cheong Wenkai', 'M', 'keefe02@gmail.com', '2002-05-28', 'C042'),

('0069', 'Herman Chew Tian Xiang', 'M', 'herman@hotmail.com', '1994-09-01', 'C042'),

('0070', 'Chua Hao Zhong', 'M', 'hz@ymail.com', '1989-01-31', 'C043')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0001', '84415722')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0002', '97480364')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0003', '95488164')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0004', '89619408')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0005', '80362832')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0006', '97046820')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0007', '83429275')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0008', '87075165')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0009', '91047645')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0010', '99484310')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0011', '91067340')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0012', '92032298')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0013', '81140038')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0014', '97722185')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0015', '93956501')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0016', '84793712')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0017', '86658394')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0018', '92144560')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0019', '97805961')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0020', '90370385')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0021', '89305430')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0022', '87157201')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0023', '94761422')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0024', '85972293')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0025', '96564338')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0026', '97520076')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0027', '86062977')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0028', '81167368')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0029', '92155152')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0030', '92786228')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0031', '95822369')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0032', '83211735')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0033', '84013799')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0034', '91463582')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0035', '96730239')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0036', '83543932')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0037', '88314696')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0038', '96583045')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0039', '99872035')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0040', '89043523')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0041', '96174194')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0042', '83870930')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0043', '87350707')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0044', '82245495')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0045', '94435335')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0046', '86183037')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0047', '83995752')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0048', '93673171')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0049', '94998605')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0050', '82753381')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0051', '91699004')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0052', '93244823')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0053', '85713618')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0054', '89800196')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0055', '81301472')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0056', '88012536')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0057', '91096609')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0058', '91699004')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0059', '94647841')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0060', '98204886')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0061', '82579834')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0062', '83408077')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0063', '96891099')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0064', '92535438')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0065', '87596988')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0066', '95776082')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0067', '83580454')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0068', '96221848')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0069', '89805557')

INSERT INTO ContactNumber(PgrID, PgrContactNo) VALUES ('0070', '90453356')

INSERT INTO Booking(BookingID, NoOfAdultTicket, AdultSalesPrice, NoOfChildTicket, ChildSalesPrice, BookDateTime, BookStatus, EventID, SessionNo, PgrID)

VALUES ('B001',1,Null,1,Null,'2021-01-15 08:00:00','Booked','E001','1','0001'),

('B002',1,Null,0,Null,'2021-01-15 08:00:00','Booked','E001','2','0003'),

('B003',1,Null,0,5,'2021-01-15 08:00:00','Booked','E002','1','0004'),

('B004',1,10,1,5,'2021-01-15 08:00:00','Booked','E002','2','0005'),

('B005',1,Null,1,Null,'2021-01-15 08:00:00','Booked','E003','1','0008'),

('B006',1,Null,0,Null,'2021-01-15 08:00:00','Booked','E003','2','0010'),

('B007',1,Null,0,Null,'2021-01-15 08:00:00','Booked','E003','3','0011'),

('B008',1,Null,2,Null,'2021-01-15 08:00:00','Booked','E004','1','0012'),

('B009',2,Null,1,Null,'2021-01-15 08:00:00','Booked','E004','2','0014'),

('B010',1,Null,1,Null,'2021-01-15 08:01:00','Booked','E004','3','0017'),

('B011',1,Null,0,Null,'2021-01-15 08:01:00','Booked','E005','1','0019'),

('B012',2,Null,0,Null,'2021-01-15 08:01:00','Booked','E005','2','0020'),

('B013',2,Null,0,Null,'2021-01-15 08:01:00','Booked','E005','3','0023'),

('B014',1,Null,1,Null,'2021-01-15 08:01:00','Booked','E006','1','0025'),

('B015',1,Null,1,Null,'2021-01-15 08:01:00','Booked','E006','2','0027'),

('B016',1,Null,0,Null,'2021-01-15 08:01:00','Booked','E007','1','0029'),

('B017',1,Null,0,Null,'2021-01-15 08:01:00','Booked','E007','2','0032'),

('B018',1,5,1,Null,'2021-01-15 08:01:00','Booked','E008','1','0034'),

('B019',1,5,1,Null,'2021-01-15 08:01:00','Booked','E008','2','0036'),

('B020',2,Null,0,Null,'2021-01-15 08:01:00','Booked','E009','1','0038'),

('B021',1,Null,0,Null,'2021-01-15 08:01:00','Booked','E009','2','0040'),

('B022',1,Null,0,Null,'2021-01-15 08:01:00','Booked','E009','3','0041'),

('B023',1,Null,0,Null,'2021-01-15 08:05:00','Booked','E010','1','0043'),

('B024',2,Null,0,Null,'2021-01-15 08:05:00','Booked','E010','2','0045'),

('B025',1,Null,0,Null,'2021-01-15 08:05:00','Booked','E010','3','0046'),

('B026',2,Null,0,Null,'2021-01-15 08:05:00','Booked','E011','1','0048'),

('B027',1,Null,0,Null,'2021-01-15 08:05:00','Booked','E011','2','0049'),

('B028',1,10,0,Null,'2021-01-15 08:05:00','Booked','E012','1','0050'),

('B029',1,10,0,Null,'2021-01-15 08:05:00','Booked','E012','2','0051'),

('B030',1,Null,1,Null,'2021-01-15 08:05:00','Booked','E013','1','0054'),

('B031',1,Null,0,Null,'2021-01-15 08:05:00','Booked','E013','2','0055'),

('B032',1,Null,0,Null,'2021-01-15 08:05:00','Booked','E014','1','0056'),

('B033',1,Null,0,Null,'2021-01-15 08:05:00','Booked','E014','2','0057'),

('B034',1,Null,0,Null,'2021-01-15 08:05:00','Booked','E015','1','0058'),

('B035',1,Null,0,Null,'2021-01-15 08:05:00','Booked','E015','2','0059'),

('B036',1,Null,0,Null,'2021-01-15 08:05:00','Booked','E015','3','0060'),

('B037',1,Null,0,Null,'2021-01-15 08:05:00','Booked','E015','4','0061'),

('B038',1,Null,0,Null,'2021-01-15 08:05:00','Booked','E016','1','0062'),

('B039',1,Null,0,Null,'2021-01-15 08:05:00','Booked','E016','2','0064'),

('B040',1,20,0,Null,'2021-01-15 08:05:00','Booked','E017','1','0066'),

('B041',2,Null,0,Null,'2021-01-15 08:05:00','Booked','E018','1','0068'),

('B042',1,Null,1,Null,'2021-01-15 08:05:00','Booked','E018','2','0070'),

('B043',1,Null,1,Null,'2021-01-15 08:05:00','Booked','E019','1','0001'),

('B044',1,Null,0,Null,'2021-01-15 08:05:00','Booked','E019','2','0003'),

('B045',1,Null,0,Null,'2021-01-15 08:05:00','Booked','E020','1','0004'),

('B046',1,Null,1,Null,'2021-01-15 08:05:00','Booked','E020','2','0005'),

('B047',1,Null,1,Null,'2021-01-15 08:05:00','Booked','E021','1','0008'),

('B048',1,Null,0,Null,'2021-01-15 08:05:00','Booked','E021','2','0010'),

('B049',1,Null,0,Null,'2021-01-15 08:05:00','Booked','E022','1','0011'),

('B050',1,Null,2,Null,'2021-01-15 08:05:00','Booked','E022','2','0012'),

('B051',2,Null,1,Null,'2021-01-15 08:05:00','Booked','E023','1','0014'),

('B052',1,Null,1,Null,'2021-01-15 08:10:00','Booked','E023','2','0017'),

('B053',1,Null,0,Null,'2021-01-15 08:10:00','Booked','E024','1','0019'),

('B054',2,Null,1,Null,'2021-01-15 08:10:00','Booked','E024','2','0020'),

('B055',2,Null,0,Null,'2021-01-15 08:10:00','Booked','E025','1','0023'),

('B056',1,Null,1,Null,'2021-01-15 08:10:00','Booked','E025','2','0025'),

('B057',1,Null,1,Null,'2021-01-15 08:10:00','Booked','E026','1','0027'),

('B058',1,Null,2,Null,'2021-01-15 08:10:00','Booked','E026','2','0029'),

('B059',1,Null,1,Null,'2021-01-15 08:10:00','Booked','E026','3','0032'),

('B060',1,Null,1,Null,'2021-01-15 08:10:00','Booked','E027','1','0034'),

('B061',1,Null,1,Null,'2021-01-15 08:10:00','Booked','E027','2','0036'),

('B062',2,Null,0,Null,'2021-01-15 08:10:00','Booked','E028','1','0038'),

('B063',1,Null,0,Null,'2021-01-15 08:10:00','Booked','E029','1','0040'),

('B065',1,Null,0,Null,'2021-01-15 08:10:00','Booked','E029','2','0041'),

('B066',1,Null,0,Null,'2021-01-15 08:10:00','Booked','E030','1','0043'),

('B067',2,Null,0,Null,'2021-01-15 08:15:00','Booked','E030','2','0045'),

('B068',1,Null,0,Null,'2021-01-15 08:15:00','Booked','E030','3','0046'),

('B069',2,16,0,Null,'2021-01-15 08:15:00','Booked','E031','1','0048'),

('B070',1,8,1,Null,'2021-01-15 08:15:00','Booked','E031','2','0049'),

('B071',1,8,0,Null,'2021-01-15 08:15:00','Booked','E031','3','0050'),

('B072',1,8,0,Null,'2021-01-15 08:15:00','Booked','E031','4','0051'),

('B073',1,8,1,Null,'2021-01-15 08:15:00','Booked','E031','5','0054')

INSERT INTO Eatery(EatyID,EatyName,EatyClHr,EatyOpHr,EatyLoc,EatyCapacity)

values ('ET01','The Caribbean Dining','23:00:00','05:00:00','Restaurant',200),

('ET02','The Big Shark','23:30:00','05:30:00','Restaurant',150),

('ET03','The Caramel Ship','23:45:00','05:45:00','Restaurant',100),

('ET04','Cottage Dining','23:59:59','06:00:00','Restaurant',100),

('ET05','East Wing','23:45:00','05:45:00','Dining Hall',100),

('ET06','West Wing','23:59:59','06:00:00','Dining Hall',100)

Insert Into Reservation(ReservID,ReservDateTime,RequiredDateTime,NoOfPax,ReservStatus,PgrID,EatyID)

Values ('0001','2022-01-15 08:00:00','2022-01-15 19:00:00','2','Booked','0001','ET01'),

('0002','2022-01-15 08:01:00','2022-01-15 19:00:00','1','Booked','0003','ET01'),

('0003','2022-01-15 08:02:00','2022-01-15 19:00:00','1','Booked','0004','ET01'),

('0004','2022-01-15 08:03:00','2022-01-15 19:00:00','4','Booked','0005','ET01'),

('0005','2022-01-15 08:04:00','2022-01-15 19:00:00','2','Booked','0008','ET01'),

('0006','2022-01-15 08:05:00','2022-01-15 19:00:00','1','Booked','0010','ET01'),

('0007','2022-01-15 08:06:00','2022-01-15 19:00:00','1','Booked','0011','ET01'),

('0008','2022-01-15 08:07:00','2022-01-15 19:00:00','3','Booked','0012','ET01'),

('0009','2022-01-15 08:08:00','2022-01-15 19:00:00','3','Booked','0014','ET01'),

('0010','2022-01-15 08:09:00','2022-01-15 19:00:00','2','Booked','0017','ET01'),

('0011','2022-01-15 08:10:00','2022-01-15 19:15:00','1','Booked','0019','ET01'),

('0012','2022-01-15 08:11:00','2022-01-15 19:15:00','3','Booked','0020','ET02'),

('0013','2022-01-15 09:00:00','2022-01-15 19:15:00','2','Booked','0023','ET02'),

('0014','2022-01-15 09:01:00','2022-01-15 19:15:00','2','Booked','0025','ET02'),

('0015','2022-01-15 09:02:00','2022-01-15 19:15:00','2','Booked','0027','ET02'),

('0016','2022-01-15 09:03:00','2022-01-15 19:15:00','3','Booked','0029','ET02'),

('0017','2022-01-15 09:04:00','2022-01-15 19:15:00','2','Booked','0032','ET02'),

('0018','2022-01-15 09:05:00','2022-01-15 19:15:00','2','Booked','0034','ET02'),

('0019','2022-01-15 09:06:00','2022-01-15 19:15:00','2','Booked','0036','ET02'),

('0020','2022-01-15 09:07:00','2022-01-15 19:15:00','2','Booked','0038','ET02'),

('0021','2022-01-15 09:08:00','2022-01-15 19:30:00','1','Booked','0040','ET02'),

('0022','2022-01-15 09:09:00','2022-01-15 19:30:00','1','Booked','0041','ET03'),

('0023','2022-01-15 09:10:00','2022-01-15 19:30:00','1','Booked','0042','ET03'),

('0024','2022-01-15 09:11:00','2022-01-15 19:30:00','2','Booked','0043','ET03'),

('0025','2022-01-15 10:00:00','2022-01-15 19:30:00','1','Booked','0045','ET03'),

('0026','2022-01-15 10:01:00','2022-01-15 19:30:00','2','Booked','0046','ET03'),

('0027','2022-01-15 10:02:00','2022-01-15 19:30:00','1','Booked','0048','ET03'),

('0028','2022-01-15 10:03:00','2022-01-15 19:30:00','1','Booked','0049','ET03'),

('0029','2022-01-15 10:04:00','2022-01-15 19:30:00','1','Booked','0050','ET03'),

('0030','2022-01-15 10:05:00','2022-01-15 19:30:00','2','Booked','0051','ET03'),

('0031','2022-01-15 10:06:00','2022-01-15 19:45:00','1','Booked','0054','ET03'),

('0032','2022-01-15 10:07:00','2022-01-15 19:45:00','1','Booked','0055','ET04'),

('0033','2022-01-15 10:08:00','2022-01-15 19:45:00','1','Booked','0056','ET04'),

('0034','2022-01-15 10:09:00','2022-01-15 19:45:00','1','Booked','0057','ET04'),

('0035','2022-01-15 10:10:00','2022-01-15 19:45:00','1','Booked','0058','ET04'),

('0036','2022-01-15 10:11:00','2022-01-15 20:00:00','1','Booked','0059','ET04'),

('0037','2022-01-15 11:00:00','2022-01-15 20:00:00','1','Booked','0060','ET04'),

('0038','2022-01-15 11:01:00','2022-01-15 20:00:00','1','Booked','0061','ET04'),

('0039','2022-01-15 11:02:00','2022-01-15 20:00:00','1','Booked','0062','ET04'),

('0040','2022-01-15 11:03:00','2022-01-15 20:00:00','2','Booked','0064','ET04'),

('0041','2022-01-15 11:04:00','2022-01-15 20:00:00','2','Booked','0066','ET04'),

('0042','2022-01-15 11:05:00','2022-01-15 20:00:00','2','Booked','0068','ET04'),

('0043','2022-01-15 11:06:00','2022-01-15 20:00:00','1','Booked','0050','ET04'),

('0044','2022-01-15 09:09:00','2022-01-15 19:30:00','1','Booked','0051','ET05'),

('0045','2022-01-15 09:10:00','2022-01-15 19:30:00','1','Booked','0052','ET05'),

('0046','2022-01-15 09:11:00','2022-01-15 19:30:00','1','Booked','0053','ET05'),

('0047','2022-01-15 10:00:00','2022-01-15 19:30:00','1','Booked','0054','ET05'),

('0048','2022-01-15 10:01:00','2022-01-15 19:30:00','1','Booked','0055','ET05'),

('0049','2022-01-15 10:02:00','2022-01-15 19:30:00','1','Booked','0056','ET05'),

('0050','2022-01-16 10:03:00','2022-01-15 19:30:00','1','Booked','0057','ET05'),

('0051','2022-01-16 10:04:00','2022-01-15 19:30:00','1','Booked','0058','ET05'),

('0052','2022-01-16 10:05:00','2022-01-15 19:30:00','1','Booked','0059','ET05'),

('0053','2022-01-16 10:06:00','2022-01-15 19:45:00','1','Booked','0060','ET05'),

('0054','2022-01-17 09:09:00','2022-01-15 19:30:00','1','Booked','0061','ET05'),

('0055','2022-01-17 09:10:00','2022-01-15 19:30:00','1','Booked','0062','ET05'),

('0056','2022-01-15 09:11:00','2022-01-15 19:30:00','2','Booked','0063','ET06'),

('0057','2022-01-16 10:00:00','2022-01-15 19:30:00','1','Booked','0064','ET06'),

('0058','2022-01-16 10:01:00','2022-01-15 19:30:00','1','Booked','0065','ET06'),

('0059','2022-01-16 10:02:00','2022-01-15 19:30:00','1','Booked','0066','ET06'),

('0060','2022-01-16 10:03:00','2022-01-15 19:30:00','1','Booked','0067','ET06'),

('0061','2022-01-17 10:04:00','2022-01-15 19:30:00','1','Booked','0068','ET06'),

('0062','2022-01-17 10:05:00','2022-01-15 19:30:00','1','Booked','0069','ET06'),

('0063','2022-01-17 10:06:00','2022-01-15 19:45:00','1','Booked','0070','ET06')

Insert Into Cuisine(CuisineID,CuisineName)

Values ('CI01','Western Cuisine'),

('CI02','Japanese Cuisine'),

('CI03','Chinese Cuisine'),

('CI04','Mexican Cuisine'),

('CI05','Middle Eastern Cuisine'),

('CI06','Indian Cuisine'),

('CI07','Italian Cuisine'),

('CI08','Greek Cuisine'),

('CI09','South-East Asia Cuisine')

INSERT INTO Dish(DishID,DishName,DishDescr,EatyID, CuisineID)

values ('D101','Beef Kway Teow','Beef slice topped on Kuay Teow with a rich beef broth','ET01', 'CI03'),

('D102','Chocolate Souffle','Light and fluffy chocolate cake','ET01','CI07'),

('D103','Bak Chor Mee Pok','Ground pork and meatball and liver topped on Mee Pok','ET01','CI03'),

('D104','Ban Mian','Ground pork balls on noodles','ET01','CI03'),

('D105','Char Kway Teow','Kuay Teow fried with sauce, and seafood','ET01','CI03'),

('D106','Hainanese Curry Rice','Curry Rice with vegetables and egg','ET01','CI03'),

('D107','Bak Kut Teh','Pork Bone soup','ET01','CI03'),

('D108','Vegetarian Char Siu Rice','Slices of mock char siu, cucumbers, white rice and is drenched in sweet gravy or drizzled with dark soy sauce','ET01','CI03'),

('D109','Duck Rice','Duck served on rice','ET01','CI03'),

('D110','Sliced Fish Soup','Sliced fish served with a milk based soup','ET01','CI03'),

('D111','Kaya Toast','Toasted bread with kaya','ET01','CI03'),

('D112','Cheng Tng','Shaved ice with longan and sago pearls','ET01','CI09'),

('D113','Nasi Lemak','Sausage, egg and chicken topped on rice','ET01','CI09'),

('D114','Pho','Soup made of beef or chicken broth and rice noodles','ET01','CI09'),

('D115','Masala Dosa','South Indian breakfast where a crispy crepe made of fermented rice and lentil batter is served with flavorful spiced potato','ET01','CI06'),

('D201','Crab Beehoon','Shredded crab served on a egg fried bee hoon','ET02','CI03'),

('D202','Rain Drop Cake','Cake made of gelatin with a fruit','ET02','CI02'),

('D203','Hokkien Mee','Fried Prawn Noodles','ET02','CI03'),

('D204','Shredded Chicken Noodles','Shredded chicken topped on Noodles','ET02','CI03'),

('D205','Vegan Beehoon','Bee Hoon with vegetables','ET02','CI03'),

('D206','Teochew Porridge','Porridge with chicken balls and fishcake','ET01','CI03'),

('D207','Ayam Penyet','Grilled Chicken topped with curry on rice','ET01','CI09'),

('D208','Nasi Goreng','Stir fried malay styled fried rice','ET01','CI09'),

('D209','Mee Soto','Yellow Noodle with chicken broth','ET01','CI09'),

('D210','Fish Head Curry','Curry vegetables with fish head','ET01','CI09'),

('D211','Roti John','Flat bread with minced meat, vegetables, chopped onions and a beaten egg','ET02','CI06'),

('D212','Ice Kachang','Ice topped with milk, red bean and corn','ET02','CI09'),

('D213','Red Velvet Cake','Chocolate cake','ET02','CI09'),

('D214','Okonomiyaki','Noodles stir fried with egg','ET02','CI02'),

('D215','Sushi Set','Salmon Sashmi with aburi prawn sashimi','ET02','CI02'),

('D301','Mee Rebus','Noodle with spicy gravy topped with egg','ET03','CI09'),

('D302','Mee Siam','Bee Hoon served with spicy, sweet and sour light gravy','ET03','CI09'),

('D303','Black Pepper Crab','Crab cooked in black pepper sauce','ET03','CI09'),

('D304','Chili Crab','Crab cooked in chilli sauce','ET03','CI09'),

('D305','Oyster Omelette','Omelette with oysters','ET03','CI03'),

('D306','Hainanese Chicken Rice','Chicken topped on rice','ET03','CI03'),

('D307','Cereal Prawns','Prawn coated in cereal and stir-fried','ET03','CI03'),

('D308','Curry Puff','Puff pastry with egg, chicken and curry filling','ET03','CI09'),

('D309','Flat Bread with Hummus','Flat bread with Hummus spread','ET03','CI08'),

('D310','Flat Bread with Guacamole','Flat bread with Guacamole spread','ET03','CI04'),

('D311','Lasagna','Layers of Mash, and beef','ET03','CI07'),

('D312','Lamb Kebab','Lamb cubes with bell pepper','ET03','CI05'),

('D401','Katong Laksa','Noodles in a laksa broth with egg and fishcake','ET04','CI09'),

('D402','Satay Beehoon','chicken satay with Bee Hoon','ET04','CI09'),

('D403','Roasted Sea Bass','Sea Bass topped with lemon lime drizzle','ET04','CI01'),

('D404','Veal Chop','Calve meat with mash','ET04','CI01'),

('D405','Wagyu Meatballs','Ground wagyu beef seasoned with paprika, thyme and based in butter','ET04','CI01'),

('D406','Caesar Salad','Tossed salad with croutons and ceasar dressing','ET04','CI07'),

('D407','Otak-otak','Grilled otak wrapped in attap leaf','ET04','CI09'),

('D408','Beef Wellington','Beef and mushroom coated in mustard wrapped in a puff pastry','ET04','CI01'),

('D409','Spiced Turkey Skewers with Cumin Lime Yougurt','Turkey breast seasoned with paprika and topped on a cumin lime yougurt','ET04','CI08'),

('D410','Pan-Seared Scallops','Scallops pan seered with butter, garlic and rosemary','ET04','CI01'),

('D411','Lobster Risotto','Risotto with a tomato based sauce topped with lobster','ET04','CI07'),

('D412','Seared Foie Gras','Duck liver topped on crackers','ET04','CI05')

Insert Into Ingredient(IngredID,IngredName)

values ('I001','Coriander Powder'),

('I002','Duck'),

('I003','Rosemary'),

('I004','Mustard Powder'),

('I005','Salt'),

('I006','Paprika'),

('I007','Lemongrass'),

('I008','Turmeric'),

('I009','Curry Leaves'),

('I010','Basil'),

('I011','Parsley'),

('I012','Brown Sugar'),

('I013','Honey'),

('I014','Ceasar Dressing'),

('I015','Rice Vinegar'),

('I016','Vinegar'),

('I017','Mayonnaise'),

('I018','Egg'),

('I019','Fishball'),

('I020','Pork'),

('I021','Beef'),

('I022','Turkey'),

('I023','Chicken'),

('I024','Flour'),

('I025','Bacon'),

('I026','Cumin'),

('I027','Cocoa Powder'),

('I028','Raw Salmon'),

('I029','Fish Cake'),

('I030','Gelatin Strips'),

('I031','Cereal'),

('I032','Soya Sauce'),

('I033','Fish'),

('I034','Seafood'),

('I035','Baking Powder'),

('I036','Beet Root'),

('I037','Rice'),

('I038','Sausage'),

('I039','Noodles'),

('I040','Longan'),

('I041','Liver'),

('I042','Hummus'),

('I043','Yogurt'),

('I044','Tau Pok'),

('I045','Milk'),

('I046','Crackers'),

('I047','Pepper'),

('I048','Vegetable'),

('I049','Butter'),

('I050','Oyster Sauce'),

('I051','Sago Pearls'),

('I052','SingLong Chilli Crab Sauce'),

('I053','Black Pepper Sauce'),

('I054','Flat Bread'),

('I055','Fruits'),

('I056','Red Bean'),

('I057','Horse Radish'),

('I058','Thyme'),

('I059','Attap Leaf'),

('I060','Cooking Wine')

Insert Into Contain(DishID,IngredID)

Values ('D101','I001'),

('D101','I009'),

('D101','I005'),

('D101','I047'),

('D102','I035'),

('D102','I049'),

('D102','I018'),

('D102','I024'),

('D102','I055'),

('D102','I056'),

('D102','I048'),

('D102','I058'),

('D102','I030'),

('D103','I011'),

('D103','I019'),

('D103','I020'),

('D103','I039'),

('D103','I041'),

('D104','I039'),

('D104','I020'),

('D105','I039'),

('D105','I034'),

('D105','I050'),

('D106','I037'),

('D106','I048'),

('D106','I023'),

('D106','I018'),

('D106','I009'),

('D106','I008'),

('D106','I045'),

('D107','I020'),

('D107','I047'),

('D107','I005'),

('D107','I032'),

('D107','I048'),

('D108','I048'),

('D108','I037'),

('D108','I032'),

('D109','I002'),

('D109','I037'),

('D109','I048'),

('D110','I033'),

('D110','I048'),

('D110','I045'),

('D110','I039'),

('D111','I054'),

('D111','I057'),

('D112','I056'),

('D112','I040'),

('D112','I051'),

('D112','I055'),

('D113','I038'),

('D113','I037'),

('D113','I018'),

('D114','I021'),

('D114','I023'),

('D114','I037'),

('D114','I039'),

('D115','I058'),

('D115','I037'),

('D201','I052'),

('D201','I059'),

('D201','I018'),

('D201','I039'),

('D202','I018'),

('D202','I035'),

('D202','I055'),

('D202','I030'),

('D203','I034'),

('D203','I039'),

('D203','I032'),

('D204','I039'),

('D204','I023'),

('D205','I039'),

('D205','I048'),

('D206','I023'),

('D206','I029'),

('D206','I037'),

('D206','I047'),

('D207','I037'),

('D207','I023'),

('D207','I009'),

('D208','I005'),

('D208','I018'),

('D208','I034'),

('D208','I037'),

('D209','I018'),

('D209','I023'),

('D209','I039'),

('D209','I044'),

('D209','I048'),

('D210','I048'),

('D210','I033'),

('D210','I045'),

('D211','I054'),

('D211','I048'),

('D211','I018'),

('D211','I060'),

('D212','I045'),

('D212','I056'),

('D212','I058'),

('D213','I027'),

('D213','I035'),

('D213','I049'),

('D214','I018'),

('D214','I029'),

('D215','I028'),

('D215','I034'),

('D215','I037'),

('D215','I057'),

('D301','I039'),

('D301','I018'),

('D301','I048'),

('D302','I039'),

('D302','I044'),

('D302','I048'),

('D303','I053'),

('D303','I034'),

('D304','I034'),

('D304','I052'),

('D305','I034'),

('D305','I018'),

('D306','I023'),

('D306','I037'),

('D307','I059'),

('D307','I034'),

('D308','I018'),

('D308','I023'),

('D308','I024'),

('D308','I009'),

('D309','I054'),

('D309','I042'),

('D310','I054'),

('D310','I055'),

('D311','I058'),

('D311','I021'),

('D312','I026'),

('D312','I048'),

('D401','I039'),

('D401','I018'),

('D401','I009'),

('D401','I019'),

('D402','I018'),

('D402','I023'),

('D402','I039'),

('D402','I048'),

('D403','I033'),

('D403','I049'),

('D403','I055'),

('D403','I058'),

('D404','I003'),

('D404','I021'),

('D404','I048'),

('D404','I049'),

('D405','I003'),

('D405','I005'),

('D405','I006'),

('D405','I021'),

('D405','I047'),

('D405','I049'),

('D405','I058'),

('D406','I014'),

('D406','I048'),

('D406','I054'),

('D407','I007'),

('D407','I008'),

('D407','I033'),

('D407','I048'),

('D407','I059'),

('D408','I004'),

('D408','I005'),

('D408','I018'),

('D408','I021'),

('D408','I024'),

('D409','I003'),

('D409','I005'),

('D409','I006'),

('D409','I007'),

('D409','I008'),

('D409','I022'),

('D409','I026'),

('D409','I043'),

('D409','I055'),

('D410','I003'),

('D410','I034'),

('D410','I048'),

('D410','I049'),

('D411','I034'),

('D411','I037'),

('D411','I048'),

('D411','I060'),

('D412','I041'),

('D412','I046')

Insert Into FoodCategory(FcID,FcName,FcDescr)

Values ('FC01','Kosher','Food that conforms to the Jewish dietary regulations'),

('FC02','Vegetarian','Food that do not contain meat'),

('FC03','Vegan','Food that do not contain animal products'),

('FC04','Halal','Food that do not contain gelatin or pork')

Insert Into Categorised\_in(FcID,DishID)

Values ('FC01','D101'),

('FC04','D101'),

('FC01','D102'),

('FC02','D102'),

('FC03','D102'),

('FC04','D102'),

('FC01','D103'),

('FC01','D104'),

('FC01','D105'),

('FC04','D105'),

('FC01','D106'),

('FC02','D106'),

('FC04','D106'),

('FC01','D107'),

('FC01','D108'),

('FC02','D108'),

('FC03','D108'),

('FC04','D108'),

('FC01','D109'),

('FC04','D109'),

('FC01','D110'),

('FC04','D110'),

('FC01','D111'),

('FC02','D111'),

('FC04','D111'),

('FC01','D112'),

('FC02','D112'),

('FC03','D112'),

('FC04','D112'),

('FC01','D113'),

('FC04','D113'),

('FC01','D114'),

('FC01','D115'),

('FC02','D115'),

('FC03','D115'),

('FC04','D115'),

('FC01','D201'),

('FC04','D201'),

('FC01','D202'),

('FC02','D202'),

('FC01','D203'),

('FC04','D203'),

('FC01','D204'),

('FC04','D204'),

('FC01','D205'),

('FC02','D205'),

('FC03','D205'),

('FC04','D205'),

('FC01','D206'),

('FC04','D206'),

('FC01','D207'),

('FC04','D207'),

('FC01','D208'),

('FC04','D208'),

('FC01','D209'),

('FC04','D209'),

('FC04','D210'),

('FC01','D211'),

('FC04','D211'),

('FC01','D212'),

('FC02','D212'),

('FC03','D212'),

('FC04','D212'),

('FC01','D213'),

('FC02','D213'),

('FC03','D213'),

('FC04','D213'),

('FC01','D214'),

('FC04','D214'),

('FC01','D215'),

('FC04','D215'),

('FC01','D301'),

('FC01','D302'),

('FC01','D303'),

('FC04','D303'),

('FC01','D304'),

('FC04','D304'),

('FC01','D305'),

('FC04','D305'),

('FC01','D306'),

('FC04','D306'),

('FC01','D307'),

('FC04','D307'),

('FC01','D308'),

('FC04','D308'),

('FC01','D309'),

('FC02','D309'),

('FC03','D309'),

('FC04','D309'),

('FC01','D310'),

('FC02','D310'),

('FC03','D310'),

('FC04','D310'),

('FC01','D311'),

('FC04','D311'),

('FC01','D312'),

('FC04','D312'),

('FC04','D401'),

('FC01','D402'),

('FC04','D402'),

('FC01','D403'),

('FC04','D403'),

('FC01','D404'),

('FC04','D404'),

('FC04','D405'),

('FC01','D406'),

('FC03','D406'),

('FC04','D406'),

('FC01','D407'),

('FC04','D407'),

('FC01','D408'),

('FC04','D408'),

('FC04','D409'),

('FC04','D410'),

('FC01','D411'),

('FC04','D411'),

('FC01','D412'),

('FC04','D412')

Insert Into CSDish(DishID,Price)

Values ('D101','5.50'),

('D102','5.50'),

('D103','5.00'),

('D104','5.00'),

('D105','6.00'),

('D106','6.00'),

('D107','6.00'),

('D108','5.00'),

('D109','5.50'),

('D110','6.00'),

('D111','4.00'),

('D112','3.00'),

('D113','6.00'),

('D114','5.00'),

('D115','6.00'),

('D201','8.00'),

('D202','6.00'),

('D203','5.00'),

('D204','4.00'),

('D205','4.00'),

('D206','4.00'),

('D207','4.50'),

('D208','4.00'),

('D209','4.00'),

('D210','8.00'),

('D211','3.00'),

('D212','3.00'),

('D213','4.50'),

('D214','3.50'),

('D215','8.00'),

('D301','4.50'),

('D302','4.50'),

('D303','4.50'),

('D304','5.00'),

('D305','5.50'),

('D306','8.50'),

('D307','10.00'),

('D308','3.50'),

('D309','4.00'),

('D310','10.00'),

('D311','2.50'),

('D312','3.00'),

('D401','5.00'),

('D402','5.00'),

('D403','5.00'),

('D404','5.50'),

('D405','5.50'),

('D406','15.00'),

('D407','15.00'),

('D408','8.00'),

('D409','4.00'),

('D410','6.00'),

('D411','15.00'),

('D412','20.00')

Insert Into Orders(PgrID,DishID,OrderPrice,OrderQty,OrderDateTime,DeliverTo,DelDateTime)

Values('0001','D101','11','2','2021-01-15 11:00:00','C001',Null),

('0001','D102','5.50','1','2021-01-15 11:01:00','C001',Null),

('0003','D103','10','2','2021-01-15 11:02:00','C002',Null),

('0003','D104','5','1','2021-01-15 11:03:00','C002',Null),

('0003','D105','6','1','2021-01-15 11:04:00','C002',Null),

('0004','D106','12','2','2021-01-15 11:05:00','C003',Null),

('0004','D107','12','2','2021-01-15 11:06:00','C003',Null),

('0005','D108','5','1','2021-01-15 11:07:00','C004',Null),

('0005','D109','11','2','2021-01-15 11:08:00','C004',Null),

('0005','D110','6','1','2021-01-15 11:09:00','C004',Null),

('0008','D111','4','1','2021-01-15 11:10:00','C005',Null),

('0008','D112','6','2','2021-01-15 11:11:00','C005',Null),

('0010','D113','6','1','2021-01-15 11:12:00','C006',Null),

('0010','D114','5','2','2021-01-15 11:13:00','C006',Null),

('0011','D115','6','1','2021-01-15 11:14:00','C007',Null),

('0011','D201','16','2','2021-01-15 11:15:00','C007',Null),

('0012','D202','12','2','2021-01-16 11:00:00','C008',Null),

('0012','D203','5','1','2021-01-16 11:01:00','C008',Null),

('0014','D204','8','2','2021-01-16 11:02:00','C009',Null),

('0014','D205','8','2','2021-01-16 11:03:00','C009',Null),

('0017','D206','8','2','2021-01-16 11:04:00','C010',Null),

('0017','D207','9','2','2021-01-16 11:05:00','C010',Null),

('0017','D208','4','1','2021-01-16 11:06:00','C010',Null),

('0019','D209','4','1','2021-01-16 11:07:00','C011',Null),

('0019','D210','28','1','2021-01-16 11:08:00','C011',Null),

('0019','D211','3','1','2021-01-16 11:09:00','C011',Null),

('0020','D212','6','2','2021-01-16 11:10:00','C012',Null),

('0023','D213','9','2','2021-01-16 11:11:00','C013',Null),

('0025','D214','7','2','2021-01-16 11:12:00','C014',Null),

('0025','D215','16','2','2021-01-16 11:13:00','C014',Null),

('0027','D301','13.5','3','2021-01-16 11:14:00','C015',Null),

('0027','D302','13.5','3','2021-01-16 11:15:00','C015',Null),

('0029','D303','13.5','3','2021-01-17 11:00:00','C016',Null),

('0032','D304','15','3','2021-01-17 11:01:00','C017',Null),

('0032','D305','16.5','3','2021-01-17 11:02:00','C017',Null),

('0034','D306','25.5','3','2021-01-17 11:03:00','C018',Null),

('0034','D307','80','2','2021-01-17 11:04:00','C018',Null),

('0036','D308','7','2','2021-01-17 11:05:00','C019',Null),

('0036','D309','8','2','2021-01-17 11:06:00','C019',Null),

('0038','D310','30','2','2021-01-17 11:07:00','C020',Null),

('0038','D311','5','2','2021-01-17 11:08:00','C020',Null),

('0040','D312','6','2','2021-01-17 11:09:00','C021',Null),

('0040','D401','10','2','2021-01-17 11:10:00','C021',Null),

('0041','D402','10','2','2021-01-15 12:00:00','C022',Null),

('0041','D403','5','1','2021-01-15 12:01:00','C022',Null),

('0042','D404','5.50','1','2021-01-15 12:02:00','C023',Null),

('0043','D405','5.50','1','2021-01-15 12:03:00','C024',Null),

('0045','D406','28','1','2021-01-15 12:04:00','C025',Null),

('0046','D407','40','2','2021-01-15 12:05:00','C026',Null),

('0046','D408','8','2','2021-01-15 12:06:00','C026',Null),

('0048','D409','4','2','2021-01-15 12:07:00','C027',Null),

('0048','D410','6','2','2021-01-15 12:08:00','C027',Null),

('0050','D411','40','2','2021-01-15 12:09:00','C029',Null),

('0050','D412','84','3','2021-01-15 12:10:00','C029',Null),

('0051','D403','15','3','2021-01-15 12:11:00','C030',Null),

('0051','D112','9','3','2021-01-16 12:00:00','C030',Null),

('0054','D115','18','3','2021-01-16 12:01:00','C031',Null),

('0054','D201','24','3','2021-01-16 12:02:00','C031',Null),

('0055','D205','12','3','2021-01-16 12:03:00','C032',Null),

('0055','D211','12','4','2021-01-16 12:04:00','C032',Null),

('0056','D304','20','4','2021-01-16 12:05:00','C033',Null),

('0057','D303','18','4','2021-01-16 12:06:00','C034',Null),

('0058','D302','18','4','2021-01-16 12:07:02','C035',Null),

('0058','D301','18','4','2021-01-16 12:40:00','C035',Null),

('0059','D306','8.50','1','2021-01-16 12:30:00','C036',Null),

('0060','D405','5.50','1','2021-01-16 13:02:00','C037',Null),

('0060','D412','40','1','2021-01-16 13:11:00','C037',Null),

('0061','D411','20','1','2021-01-17 12:32:29','C038',Null),

('0062','D408','8','1','2021-01-17 12:33:23','C039',Null),

('0064','D311','5','2','2021-01-17 17:17:17','C040',Null),

('0064','D312','6','2','2021-01-16 16:16:16','C040',Null),

('0066','D308','7','2','2021-01-15 15:15:15','C041',Null),

('0068','D312','6','2','2021-01-17 12:13:14','C042',Null),

('0070','D411','40','2','2021-01-17 15:16:17','C043',Null),

('0070','D404','16.5','3','2021-01-17 12:19:20','C043',Null)

## SQL output for Passenger Table

SELECT \* FROM Passenger

Table

Description automatically generated

## Table Description automatically generatedSQL output for ContactNumber Table

SELECT \* FROM ContactNumber

Table

Description automatically generated with medium confidence

Table

Description automatically generated

## SQL output for Reservation Table

SELECT \* FROM Reservation

Table

Description automatically generated

Table

Description automatically generated

## SQL output for Booking Table

SELECT \* FROM Booking

Table

Description automatically generatedTable

Description automatically generated

## SQL output for Event Table

SELECT \* FROM Event

A picture containing graphical user interface

Description automatically generated

## SQL output for EventType Table

SELECT \* FROM EventType

Graphical user interface, table

Description automatically generated

## SQL output for EventSession Table

SELECT \* FROM EventSession

Table

Description automatically generated

Table

Description automatically generated

## SQL output for Eatery Table

SELECT \* FROM Eatery

Table

Description automatically generated with low confidence

## SQL output for Dish Table

SELECT \* FROM Dish

Text

Description automatically generated with medium confidenceTable

Description automatically generated

## SQL output for Ingredient Table

SELECT \* FROM Ingredient

Graphical user interface

Description automatically generated with low confidence

Table

Description automatically generated

## SQL output for Contain Table

SELECT \* FROM Contain

A picture containing text, electronics

Description automatically generated

A screenshot of a computer

Description automatically generated with low confidence

A picture containing text, electronics

Description automatically generated

A screenshot of a computer

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with low confidence

## SQL output for Cuisine Table

SELECT \* FROM Cuisine

Table

Description automatically generated

## SQL output for FoodCategory Table

SELECT \* FROM FoodCategory

Graphical user interface, text, application, chat or text message

Description automatically generated

## SQL output for Categorised\_in Table

SELECT \* FROM Categorised\_in

A screenshot of a computer

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with low confidence

Table

Description automatically generated

## SQL output for CSDish Table

SELECT \* FROM CSDish

Table

Description automatically generatedTable

Description automatically generated with medium confidence

## SQL output for Orders Table

SELECT \* FROM Orders

Table

Description automatically generated

Table

Description automatically generated

# List of Queries

## Sherlena’s Queries

1. List the passengers name and number of times ordered Vegetarian meals in ascending order from the most number of orders to the least number of orders

SELECT Passenger.PgrName, SUM(Orders.OrderQty) AS 'Number of times ordered Vegetarian'

FROM Passenger

INNER JOIN Orders

ON Passenger.PgrID = Orders.PgrID

INNER JOIN Categorised\_in

ON Categorised\_in.DishID = Orders.DishID

INNER JOIN FoodCategory

ON Categorised\_in.FcID = FoodCategory.FcID

WHERE FoodCategory.FcName = 'Vegetarian'

GROUP BY Passenger.PgrName

ORDER BY SUM(Orders.OrderQty) DESC, Passenger.PgrName ASC

Results:

Table

Description automatically generated

Reasoning behind query:

To find out if there is a demand for vegetarian options and if more dishes need to be catered.

1. List the number of children tickets purchased for the different event ordered by the popularity of booking.

SELECT Event.EventName, SUM(Booking.NoOfChildTicket) AS 'Sum of children ticket purchased'

FROM Booking

INNER JOIN EventSession

ON Booking.EventID = EventSession.EventID AND Booking.SessionNo = EventSession.SessionNo

INNER JOIN Event

ON Booking.EventID = Event.EventID

WHERE Booking.NoOfChildTicket IN (SELECT NoOfChildTicket FROM Booking WHERE NoOfChildTicket >= 1)

GROUP BY Event.EventName

ORDER BY SUM(Booking.NoOfChildTicket) DESC, Event.EventName ASC

Results:

Graphical user interface, table

Description automatically generated

Reasoning behind query:

To find out which activity is popular among children and if more event sessions for a event should be opened.

1. List the dishes in each eatery with at least 5 orders.

SELECT Eatery.EatyName, Dish.DishName, SUM(Orders.OrderQty) AS 'Number of Orders'

FROM Orders

INNER JOIN Dish

ON Orders.DishID = Dish.DishID

INNER JOIN Eatery

ON Dish.EatyID = Eatery.EatyID

GROUP BY Dish.DishName, Eatery.EatyName

HAVING SUM(Orders.OrderQty) >= 5

ORDER BY Eatery.EatyName ASC, SUM(Orders.OrderQty) DESC, Dish.DishName ASC

Results:

Table

Description automatically generated

Reasoning behind query:

To find out the dishes in the eatery that has at least 5 orders. This is to gage the popularity of the dishes to quickly cater to the demands.

## Vincent’s Queries

1. List the popularity of the eateries in descending order by the amount of food ordered by the passengers

Select e.EatyName ,Count(OrderQty) as 'Amt of Food Ordered'

From Eatery e inner join Dish d on e.EatyID = d.EatyID

inner join Orders o on d.DishID = o.DishID

Group By e.EatyName

Order By 'Amt of Food Ordered' desc

Results:

Table

Description automatically generated

Resoning behind query:

To find out the most popular Eatery on the cruise which allows the other Eateries to improve on their dishes

1. List the popularity of the event type for adult passengers in descending order

Select et.ETName , Count(NoOfAdultTicket) as 'Amt of Adult bookings'

From EventType et inner join Event e on et.ETID =e.ETID

inner join EventSession es on e.EventID = es.EventID

inner join Booking b on e.EventID = es.EventID

Group By ETName

Order by Count(NoOfAdultTicket) desc

Resoning behind query:

To find out which Category of events do the adults like. This can be used as a guide for the upcoming trips for passengers and for the most popular events to increase their sessions to cater for more bookings

Results:

Table

Description automatically generated

1. List the dishes that has the least amount of orders

Select DishName , Sum(OrderQty) as 'Number of orders'

From Dish d inner join CSDish cs on d.DishID = cs.DishID

inner join Orders o on o.DishID = d.DishID

group by dishName

having sum(OrderQty)=1

order by SUm(OrderQty) asc

Resoning behind query:

To find the dishes that are not popular among the passengers, so that the various eateries would be able to change their menu to get more sales

Table

Description automatically generated