HOLA

SEMESTER 1 Continuous Assessment 4
Normalisation and physical design

Ollscoil Chathair
Bhaile Átha Cliath
Dublin City University

MODULE: CA218 - Introduction to Databases

PROGRAMME(S): **CASE2**

YEAR OF STUDY: 2

EXAMINER(S):

Yalemisew Abgaz Ext:6714

INSTRUCTIONS: **This template is provided for completing the questions in CA4. This template must be used and any modification should be explicitly indicated.**
Follow the instructions provided in the Loop page under the Assessments section.
This is a group work of two students.

ID: (Enter your DCU Student ID number in the right box below)

Student 1

2	0	4	1	8	7	1	2

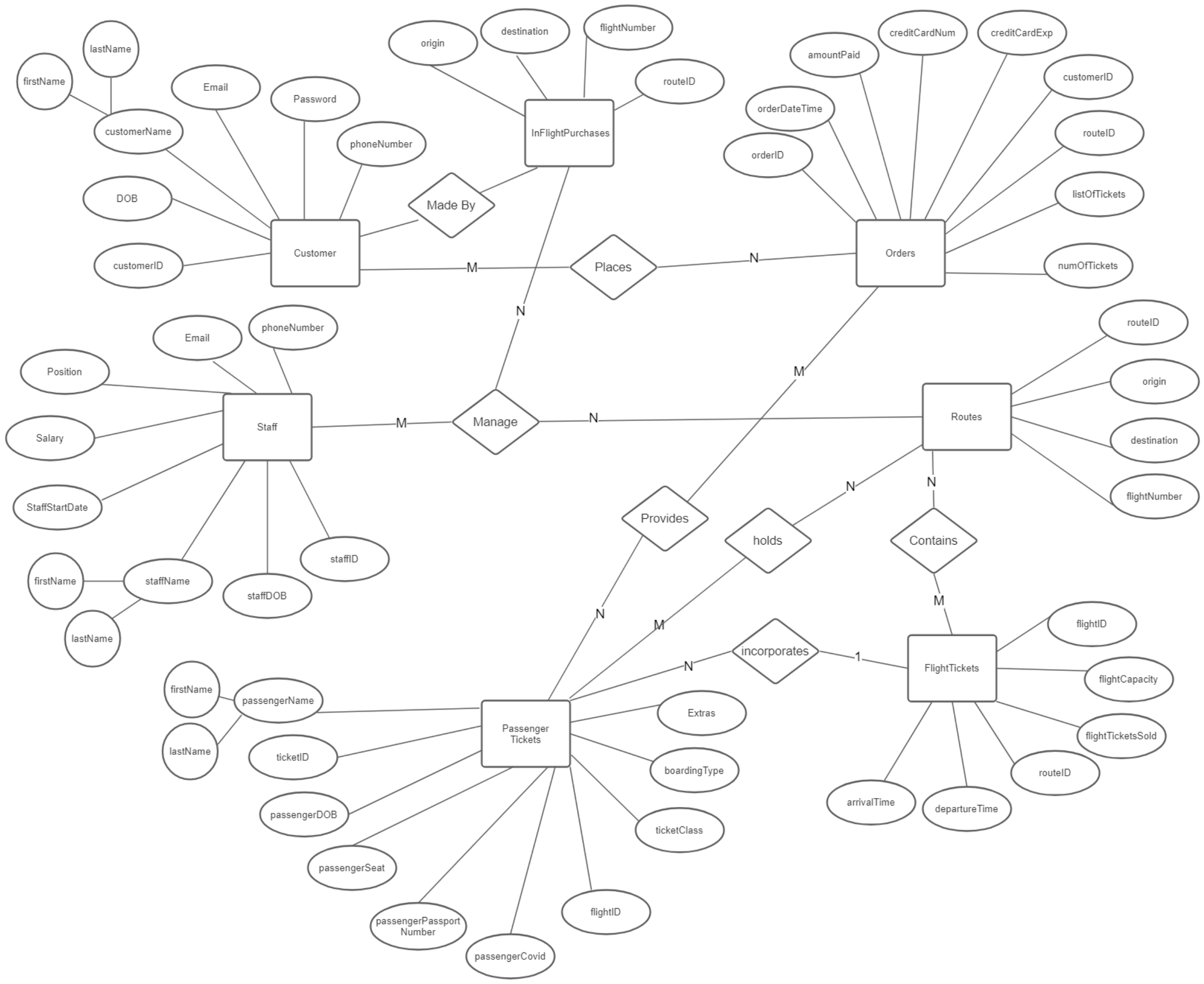
Student 2

2	0	4	1	0	9	7	4

Group Number: 50

1. **Show your final ER diagram from CA2 (5 Marks).**
   Following feedback from CA2 the following revisions are made.[1]

   1. Changed one of the relationships from contains to holds as we had to relations called contains.
   2. Added a new relationship “incorporates” with mapping N:1.
   3.



[1] The date of birth was represented as multivalued wrongly, such issues need to be fixed in your revised diagram. The second change is made to fix some of the typos, the third change is made for the purpose of demonstrating how to include revisions on your ER diagram and to show how to model 1:N relationships.

2. Map your ER diagram into a logical table following the ER mapping steps (25 Marks).

Step 1: Map Strong Entities

Customer (customerID, customerName, customerDOB, email, password, customerPhoneNumber)
Orders (orderID, orderDateTime, amountPaid, creditCardNum, creditCardExp, customerID, listOfTickets, numOfTickets)
PassengerTickets (ticketID, passengerName, passengerDOB, passengerSeat, passengerPassportNumber, passengerCovid, flightID, boardingType, ticketClass, extras)
Routes (routeID, origin, destination, flightNumber)
Staff (staffID, staffName, staffDOB, staffStartDate, staffPosition, staffSalary, staffEmail, staffPhoneNumber)
InFlightPurchases (inFlightID, ItemsPurchased, ticketID, amountPaid)
FlightTickets (flightID, flightCapacity, flightTicketsSold, routeID, departureTime, arrivalTime)

Step 2: Map Weak Entities

The above ER diagram does not have any weak entity.

Step 3: Map 1:1 Relationships

The above ER diagram does not have any 1:1 relationship.

Step 4: Map 1:N/N:1 Relationships

PassengerTickets (ticketID, passengerName, passengerDOB, passengerSeat, passengerPassportNumber, passengerCovid, flightID, boardingType, ticketClass, extras)
FK: flightID

Step 5: Map M:N Relationships

Manage *(<u>staffID, routeID, inFlightID</u>)*
FK: staffID, routeID and inFlightID
Places *(<u>customerID, orderID</u>)*
FK: customerID and orderID
Provides(<u>*orderID, ticketID*</u>)
FK: orderID and ticketID
Holds(<u>*ticketID, routeID*</u>)
FK: ticketID and routeID
Contains(<u>*flightID, routeID*</u>)
FK: flightID and routeID

Step 6: Map Multivalued attributes

The above ER diagram does not have any multivalued attribute.

Final Tables:

Customer (customerID, customerName, customerDOB, email, password, customerPhoneNumber)
Orders (orderID, orderDateTime, amountPaid, creditCardNum, creditCardExp, customerID, listOfTickets, numOfTickets)
PassengerTickets (ticketID, passengerName, passengerDOB, passengerSeat, passengerPassportNumber, passengerCovid, flightID, boardingType, ticketClass, extras)
Routes (routeID, origin, destination, flightNumber)
Staff (staffID, staffName, staffDOB, staffStartDate, staffPosition, staffSalary, staffEmail, staffPhoneNumber)
InFlightPurchases (inFlightID, ItemsPurchased, ticketID, amountPaid)
FlightTickets (flightID, flightCapacity, flightTicketsSold, routeID, departureTime, arrivalTime)

3. Normalise each table up to 3rd normal form (1st Normal form, 2nd Normal form, 3rd normal form) (30 Marks). Note: you may check each table resulting from step 2 for 1st , 2nd 3rd normal form s and if not normalised, convert the table into the corresponding normal form.

   1st Normal form
   All the tables are in their 1NF. No intersection of rows and columns contain a multiple value.
   2nd Normal form
   All the tables are in their 2NF. No partial dependency is observed.
   3rd normal form
   PassengerTickets can get the route information and flight information by using the **flightID** key.
   Orders can get the customer information by using the **customerID** key.

4. Using project database available to you on MySQL, write a DDL to create the tables in your database (focus only on the core tables, only 3 tables) (30 Marks). Ensure you use the correct datatypes, length, primary key, and foreign key (if any).

   Done in Group50.sql file.

5. Write an SQL DML to Insert at least 3 records in the tables you created in Q4. (10 Marks)

   Done in Group50.sql file.