Database Homework 2

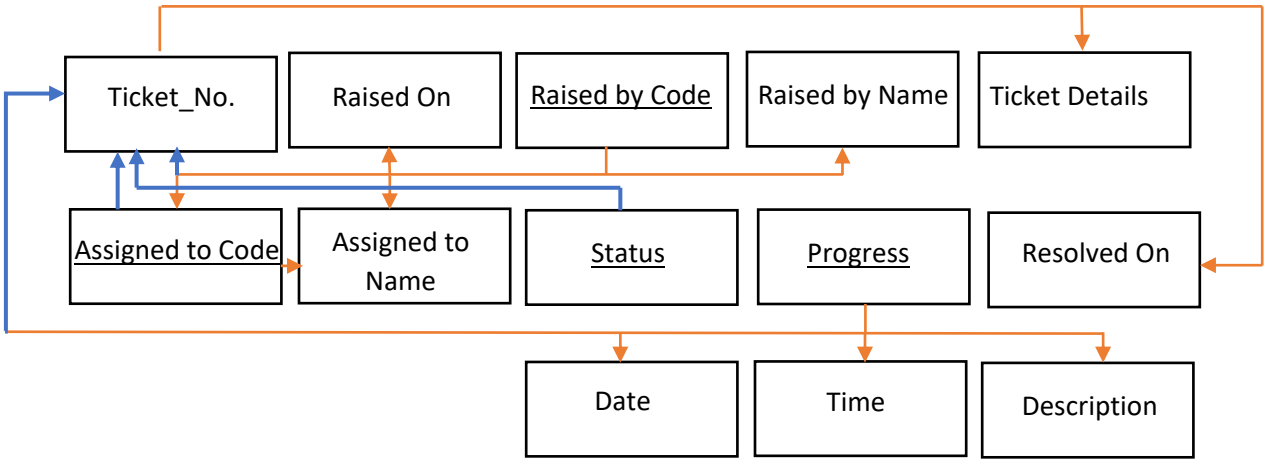
Ali Muhammad Asad

aa07190

Q1) Service Desk

From the table given in the question, the following normalization has been done;

1 NF:



The arrows in blue represent functional dependencies.

2 NF:

Tickets (Ticket No., Ticket Details, Resolved On)

Status (Status_ID,)

Raised_Info (Raised_by_Code, Raised On, Raised by Name)

Assigned_Info (Assigned_to_Code, Assigned_to_Name)

Progress (Progress_ID, Progress_info (Date, Time, Description))

Functional Dependencies (Tickets – Status, Raised_Info, Assigned_Info, Progress)

3 NF:

Tickets (Ticket No., Ticket Details, Resolved On)

Status (Status_ID,)

Raised_Info (Raised_by_Code, Raised On, Raised by Name)

Assigned_Info (Assigned_to_Code, Assigned_to_Name)

Progress (Progress_ID)

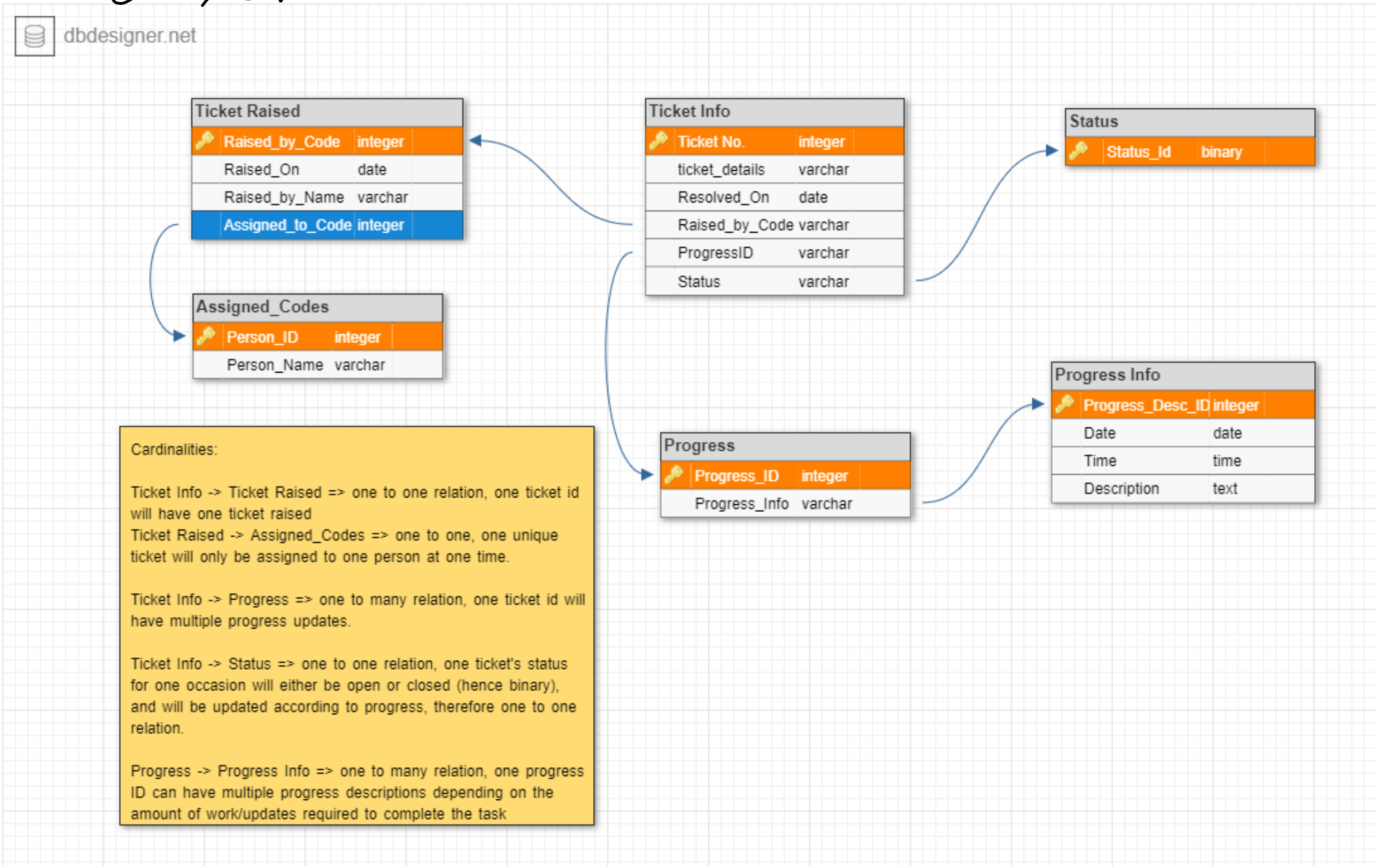
Progress_info (Progress_Desc_ID, Date, Time, Description)

Functional Dependencies (Tickets – Status, Raised_Info, Assigned_Info, Progress

Progress – Progress_info)

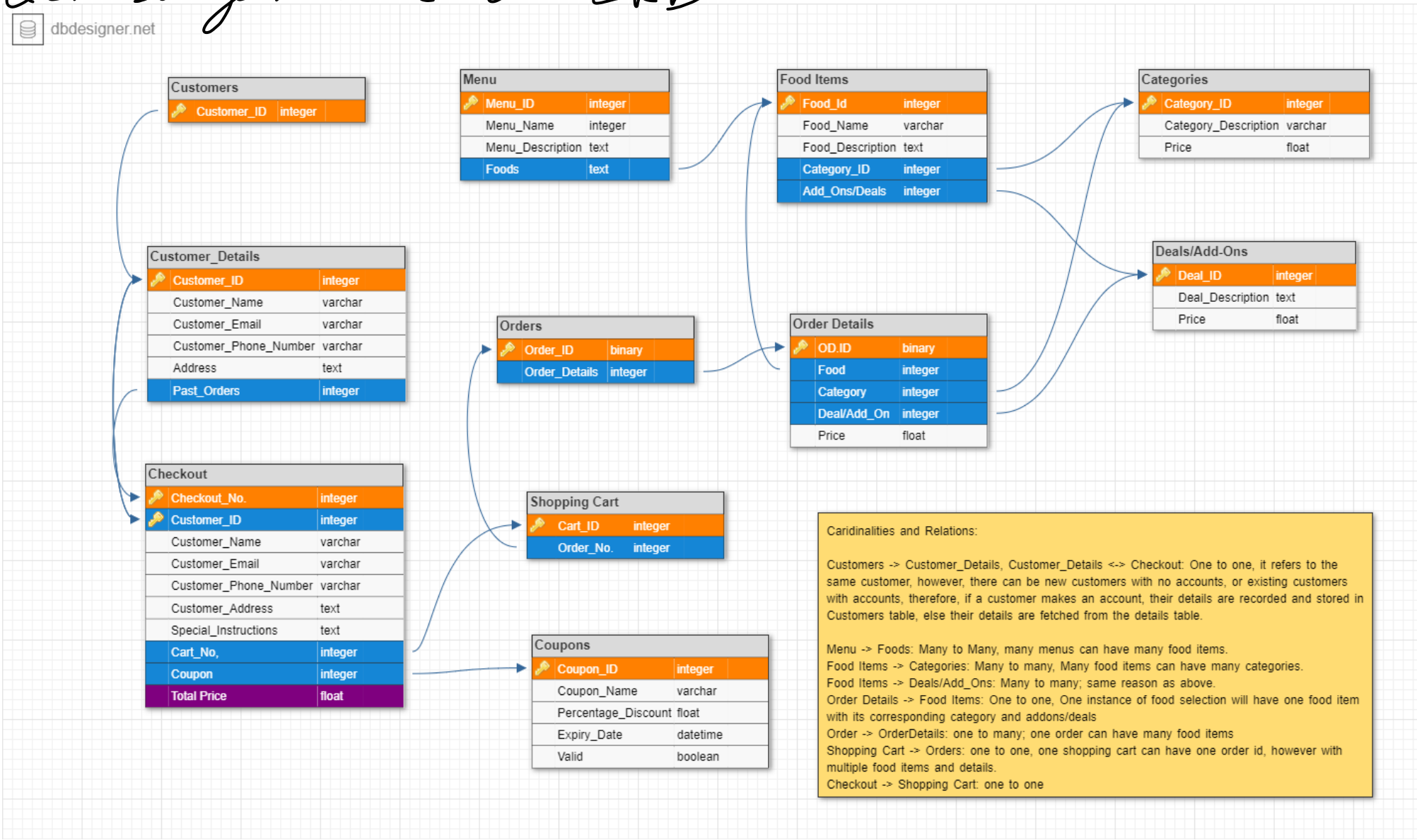
Transitive Dependencies (Progress_Info – Progress_Desc_ID, Date, Time, Description)

(Q1) ERD


dbdesigner.net
Ticket Raised
Raised_by_Code integer
Raised_On date
Raised_by_Name varchar
Assigned_to_Code integer
Assigned_Codes
Person_ID integer
Person_Name varchar
Ticket Info
Ticket No. integer
ticket_details varchar
Resolved_On date
Raised_by_Code varchar
ProgressID varchar
Status varchar
Status
Status_Id binary
Progress
Progress_ID integer
Progress_Info varchar
Progress Info
Progress_Desc_ID integer
Date date
Time time
Description text
Cardinalities:
Ticket Info -> Ticket Raised => one to one relation, one ticket id will have one ticket raised
Ticket Raised -> Assigned_Codes => one to one, one unique ticket will only be assigned to one person at one time.
Ticket Info -> Progress => one to many relation, one ticket id will have multiple progress updates.
Ticket Info -> Status => one to one relation, one ticket's status for one occasion will either be open or closed (hence binary), and will be updated according to progress, therefore one to one relation.
Progress -> Progress Info => one to many relation, one progress ID can have multiple progress descriptions depending on the amount of work/updates required to complete the task

Q2) Burger-O-Clock ERD

dbdesigner.net



Caridinalities and Relations:

Customers -> Customer_Details, Customer_Details <-> Checkout: One to one, it refers to the same customer, however, there can be new customers with no accounts, or existing customers with accounts, therefore, if a customer makes an account, their details are recorded and stored in Customers table, else their details are fetched from the details table.

Menu -> Foods: Many to Many, many menus can have many food items.
Food Items -> Categories: Many to many, Many food items can have many categories.
Food Items -> Deals/Add_Ons: Many to many; same reason as above.
Order Details -> Food Items: One to one, One instance of food selection will have one food item with its corresponding category and addons/deals
Order -> OrderDetails: one to many; one order can have many food items
Shopping Cart -> Orders: one to one, one shopping cart can have one order id, however with multiple food items and details.
Checkout -> Shopping Cart: one to one