Database Design Report

COMP23111Database Systems

Name: Zihao Chen

Account: a65914zc

Date: November 8th

TABLE OF CONTENTS

Introduction

This report will discuss three parts about the database design for Milestone One. The entity in the ERD will be directly referenced and ‘attributes’ are marked by single quotation.

Entity Relationship Diagram

The entity relationship diagram was provided in figure 1 with related analyses.

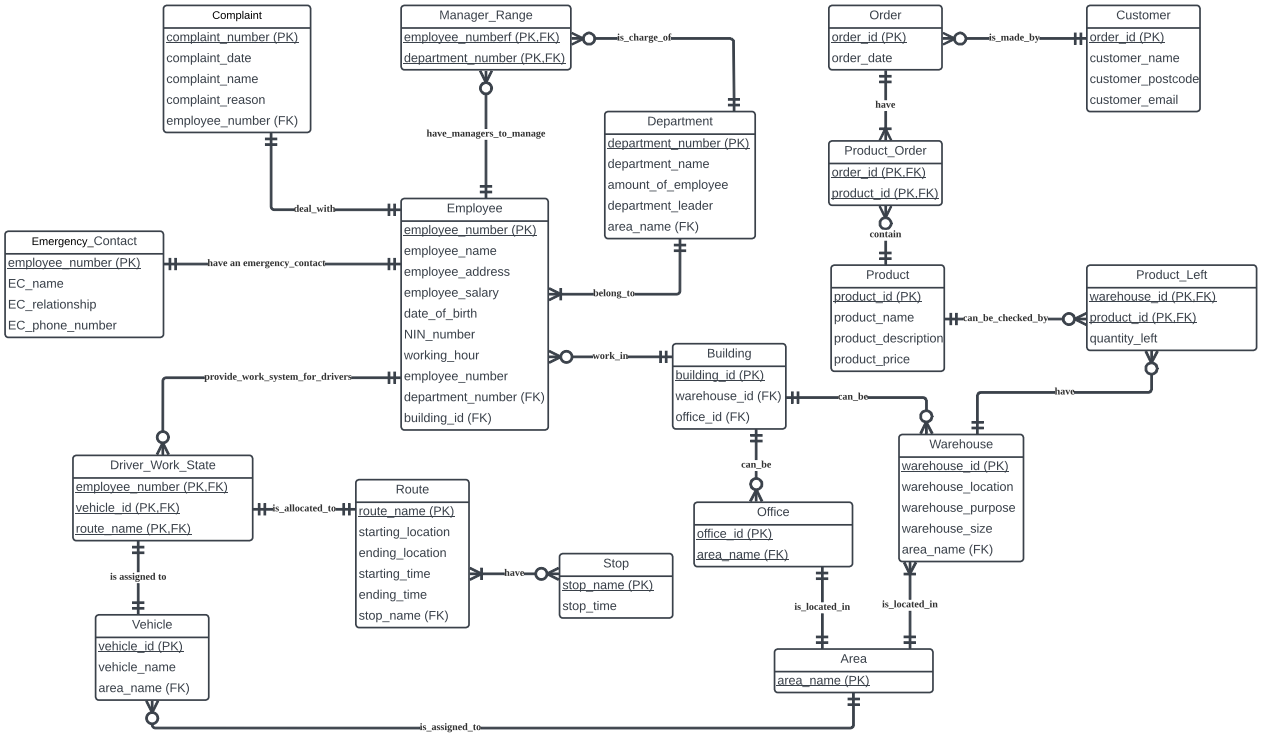

Figure 1: ER diagram

Firstly, ‘employee number’ was designed as the primary key in Employee entity, since it could identify each employee directly. This entity should belong to one Department entity for each employee and attributes were provided below such as ‘employee address’ and reported managers through ‘employee number’. Similarly, ‘department number’ was the primary key in Department entity, since different kinds of departments could be recognised through distinct ID, not affected by their amount in one specific area. And related attributes were listed such as ‘department name’ and the specific ‘area name’. One department may have one or more staffs, but one employee only belonged to one department. According to materials, each employee could leave an emergency contact,

therefore, the weak entity Emergency Contact with basic attributes like ‘name’, ‘relationship’ and ‘phone number’ would be linked to Employee entity depended on primary key ‘employee number’ one to one without its own keys.

Secondly, when the employee number belonged to the HR, the chosen staffs would deal with complaints one to one and related details were recorded like complaint’s ‘date’, ‘reason’ and ‘name of customer’. Similarly, people from the management could attend to different departments. Besides, each driver would be assigned to one company vehicle with ‘name’ and ‘unique ID’ on one selected road with details such as ‘time and location of the starting and ending site’. Sometimes there may be some stop places on the route also with ‘time recording’. The ‘vehicle id’ and ‘route name’ were viewed as the primary key of Vehicle and Route entity because of unique representativeness.

Moreover, building entity was designed to construct the relationship of employees’ allocated company building, which could be offices or warehouses. Both had the unique ‘area name’ and Warehouse entity contained ‘purpose’, ‘size’ and ‘location’. Besides, ‘warehouse id’ could directly catch the clear building information, not affected by the situation that one specific area may exist many sites. One building may include zero or more employees but one employee would only be allocated to one department building.

Finally, product entity was provided and people could check its ‘quantity left’ in the specific warehouse. Furthermore, the ‘name’, ‘description’ and ‘price’ of one product could be checked through ‘product id’, which was also the primary key because of its unique. After that, the order situation containing several products could be derived by the unique order id; however, for the product, orders are dispensable but for any order, there must be at least one product included. The weak entity Customer with attributes like ‘name’ ‘postcode’ and ‘email’ was linked to Order entity by primary key ‘order id’, making every order has the customer information without own keys. One customer could make multiple orders so the cardinality constraint was zero or many but one order could only be purchased by one customer.

Normalisation Thinking

The entity relationship diagram in the third normal form is shown in figure 2 with the normalisation thinking and reflection.

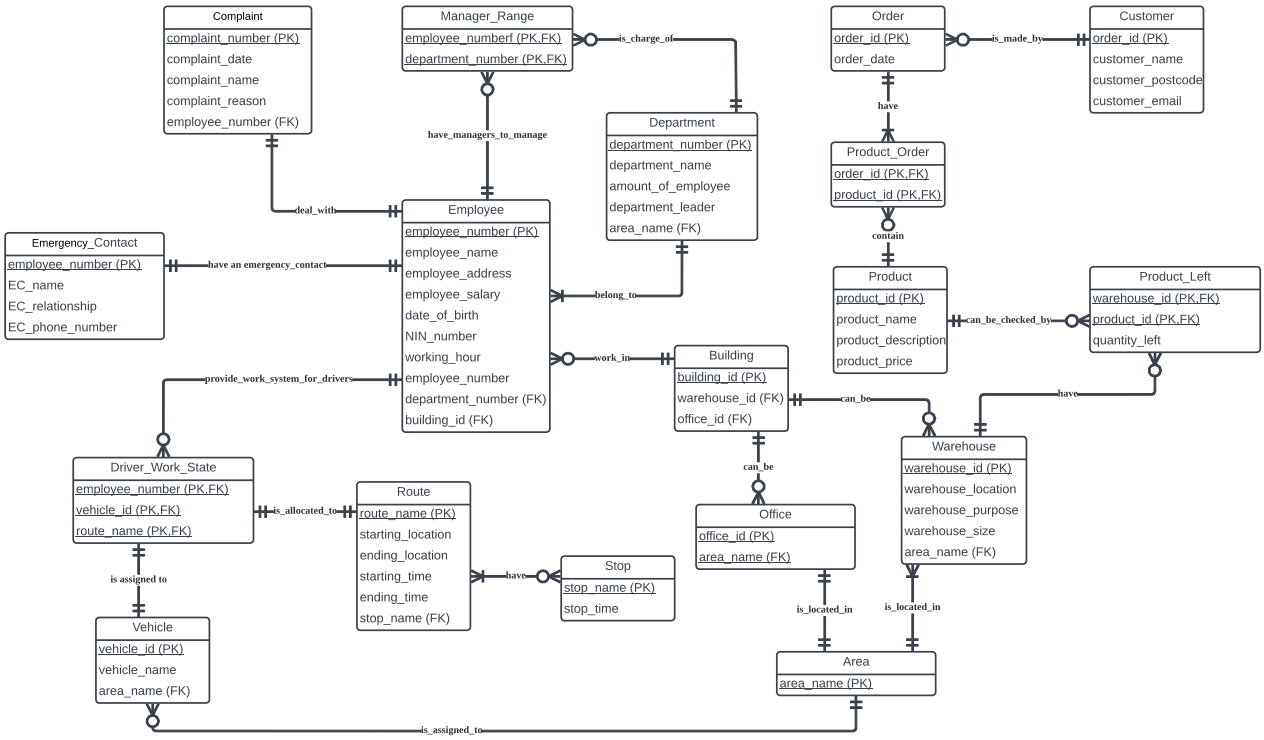

Figure 2: ER diagram in 3NF

Firstly, this diagram had conformed the first normal form because every table cell would contain one value.

Secondly, Employee' attributes like 'date of birth' or 'working hour' (for those employee number belonged to drivers) could only be checked through the primary key without functional dependence. Besides, the 'building id' and 'assigned department' were satisfied by two foreign keys linked to different entities, therefore, there was also no transitive relation and attributes of Building and Department entity were independent from 'employee number'. Similarly, for Department entity, there was a foreign key of 'area name', which was finally related to Area entity so that department locations could be searched. The composite key was designed in Manager Range entity (bridge entity) and they were isolated so the transitive dependence was rejected.

Moreover, for Complaint entity, its contents were all independent and there was a foreign key of 'employee number' to link every HR employee details, making sure there was no functional and transitive relation when HR employee was allocated to one complaint in complaint information tables. Also, attributes from weak entity Emergency Contact were totally dependent on the primary key satisfying the third normal form.

Besides, an individual environment with composite keys was designed on Driver Work State entity. The aim of this step was to make sure details of Vehicle

and Route entity would only depend on their own primary keys independent from ‘employee number’. After that, Route entity had independent attributes and one foreign key was linked to Stop entity. Therefore, ‘time recording’ in stop entity depended on its own primary key ‘stop id’ not affected by route name, following the third normal form. Vehicle entity had a foreign key linked to Area entity, making sure area names would depend on ‘vehicle id’ without transitive dependence.

Furthermore, Building entity had two foreign keys to get connections with two kind of work locations, which also make sure attributes like ‘warehouse location’ only depending on its own primary key without transitive dependence. Both Office and Warehouse entity had a foreign key linked to Area entity so that area name only depended on the location primary keys following the third normal form.

Finally, there were composite keys on Product Left entity, since the product quantity left in warehouse would be derived by both ‘warehouse id’ and ‘product id’ in one specific relationship following the third normal form. Differently, other attributes of Product entity like ‘description’ or ‘price’ fully depended on the primary key ‘product id’. For Order Product entity, there were also composite keys to connect Order entity with Product entity representing many-to-many relationship, which was independent from other attributes. ‘Order date’ fully depended on ‘order id’ and attributes of Weak entity Customer like ‘customer postcode’ totally depended on the primary key satisfying the third normal form.

Relational Scheme

According to the entity relationship diagram in 3NF, the relational scheme was provided with reasonable foreign key constraints.

Employee (employee_number, employee_name, employee_address, employee_salary, date_of_birth, NIN_number, working_hour, employee_number)

FK department_number -> Department (department_number)
FK building_id -> Building (building_id)
ON UPDATE CASCADE ON DELETE RESTRICT

Department (department_number, department_name, amount_of_employee, department_leader)

FK area_name -> Area ni(area_name)
ON UPDATE CASCADE ON DELETE RESTRICT

Manager_Range (employee_number, department_number)
FK employee_number -> Employee (employee_number)
FK department_number -> Department (department_Number)
ON UPDATE CASCADE ON DELETE SET NULL

Complaint (complaint_number, complaint_date, complaint_name, complaint_reason)
FK employee_number -> Employee (employee_number)
ON UPDATE CASCADE ON DELETE SET NULL

Emergency_Contact (employee_number, EC_name, EC_relationship, EC_phone_Number)

Driver_Work_State (employee_number, vehicle_id, route_name)
FK employee_number -> Employee (employee_number)
FK vehivle_id -> Vehicle (vehicle_id)
FK route_name -> Route (route_name)
ON UPDATE CASCADE ON DELETE SET NULL

Vehicle (vehicle_id, vehicle_name)
FK area_name -> Area (area_name)
ON UPDATE CASCADE ON DELETE RESTRICT

Route (route_name, starting_location, ending_location, starting_time, ending_time)
FK stop_name -> Stop (stop_name)
ON UPDATE CASCADE ON DELETE SET NULL

Stop (stop_name, stop_time)

Building (building_id)
FK warehouse_id -> Warehouse (warehouse_id)
FK office_id -> Office (office_id)
ON UPDATE CASCADE ON DELETE CASCADE

Office (office_id)
FK area_name -> Area (area_name)
ON UPDATE RESTRICT ON DELETE RESTRICT

Area (area_name)
Warehouse (warehouse_id, warehouse_location, warehouse_purpose, warehouse_size)
FK area_name -> Area (area_name)

ON UPDATE CASCADE ON DELETE SET NULL
Product (product_id, product_name, product_description, product_price)

Product_Left (warehouse_id, product_id, quantity_left)
FK warehouse_id -> Warehouse (warehouse_id)
FK product_id -> Product (product_id)
ON UPDATE CASCADE ON DELETE RESTRICT

Product_Order (product_id, order_id)
FK product_id -> Product (product_id)
FK order_id -> Order (order_id)
ON UPDATE RESTRICT ON DELETE RESTRICT

Order (order_id, order_date)

Customer (order_id, customer_name, customer_postcode, customer_email)

The entity relationship was converted to the required textual format and referenced foreign keys were pointed with company-based appraisal constraints.

Firstly, for Employee, UPDATE CASCADE was considered to make sure its flexibility when new employee details were added, so that contents of foreign keys could also be improved at the same time; DELETE RESTRICT was aimed to keep security of foreign keys when an employee record was tried to be deleted. Same constraints were added on Department, so it was allowed to update new records together with contents of foreign keys; however, when the department number had errors, corresponding foreign keys would not be affected because deleting operation was denied.

Besides, Manager Range and Driver Work State had same constraints. When a new record was added, related tables should update as follows through UPDATE CASCADE. But if the content was deleted, the connected position would be null or default (Innode not support) and wait the next key input through DELETE SET NULL. For example, if a new manager attended to different departments, joined tables would be created. When this manager left, the employee number part of composite keys would be null with the filled department number, waiting a new manager, which was effective for tables management.

Moreover, Product Left and Product Order were also designed through composite keys. DELETE RESTRICT was chosen because when deleting was requested, other attributes cannot be affected. For example, when sets of

products had been added in one order, customers cannot randomly change the product situation especially when the payment was confirmed, which was the common targets of SQL INJECT. Similarly, Product Order would also use UPDATE RESTRICT to keep one completed order, preventing random changes. However, Product Left used UPDATE CASCADE to make sure the flexible management when a warehouse received packages.

Furthermore, Office had to use UPDATE RESTRICT, since there would be only one office in each specific area. DELETE RESTRICT guaranteed deleting one office was not allowed because the specific area would serve other entities. UPDATE CASCADE was used in Building to create a new building in some area and pass this request on the next entity. DELETE CASCADE was considered in this bridge entity to make sure the deleting request could pass on the foreign tables.

Finally, Route and Warehouse have same constraints, because they both required to update the related tables quickly and flexibly by UPDATE CASCADE no matter applying a new warehouse in specific area or adding stop sites on a route. DELETE SET NULL was considered to make sure effective tables. For example, when a driver was assigned to a route, which may contain several routes, when the route work was done, the foreign key stop name could be set as null, then, some stop names would just be updated next time.

Conclusion

In conclusion, this report discusses the database design for Milestone One related to the diagram design, the normalization thinking and the relational scheme with foreign key constraints.