NAME: KHAIRUNNISA KHAN BINTI TALIB

TITLE: DATABASE DESIGN

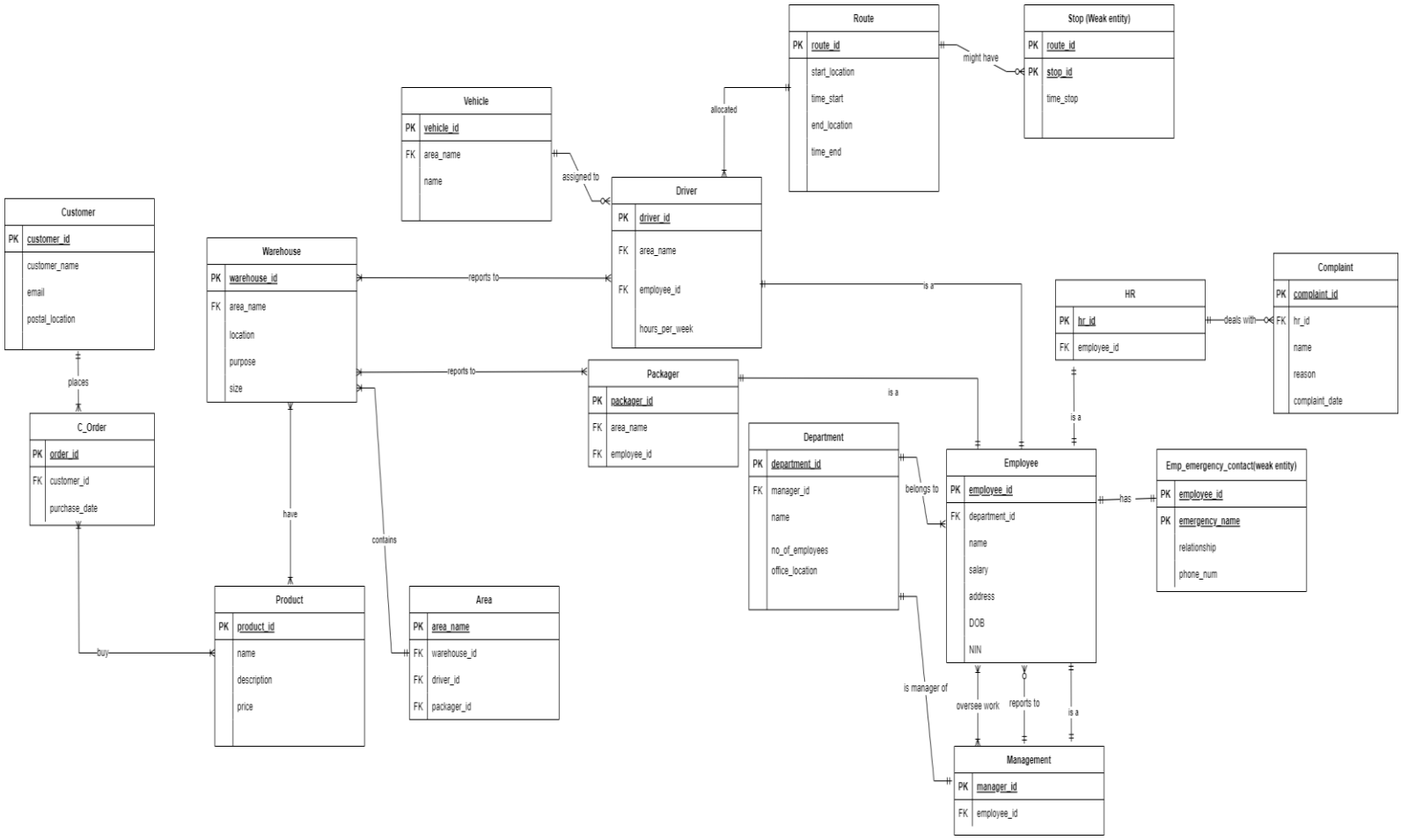
MODULE CODE: COMP23111 DATABASE SYSTEM

DATE:09/11/2022

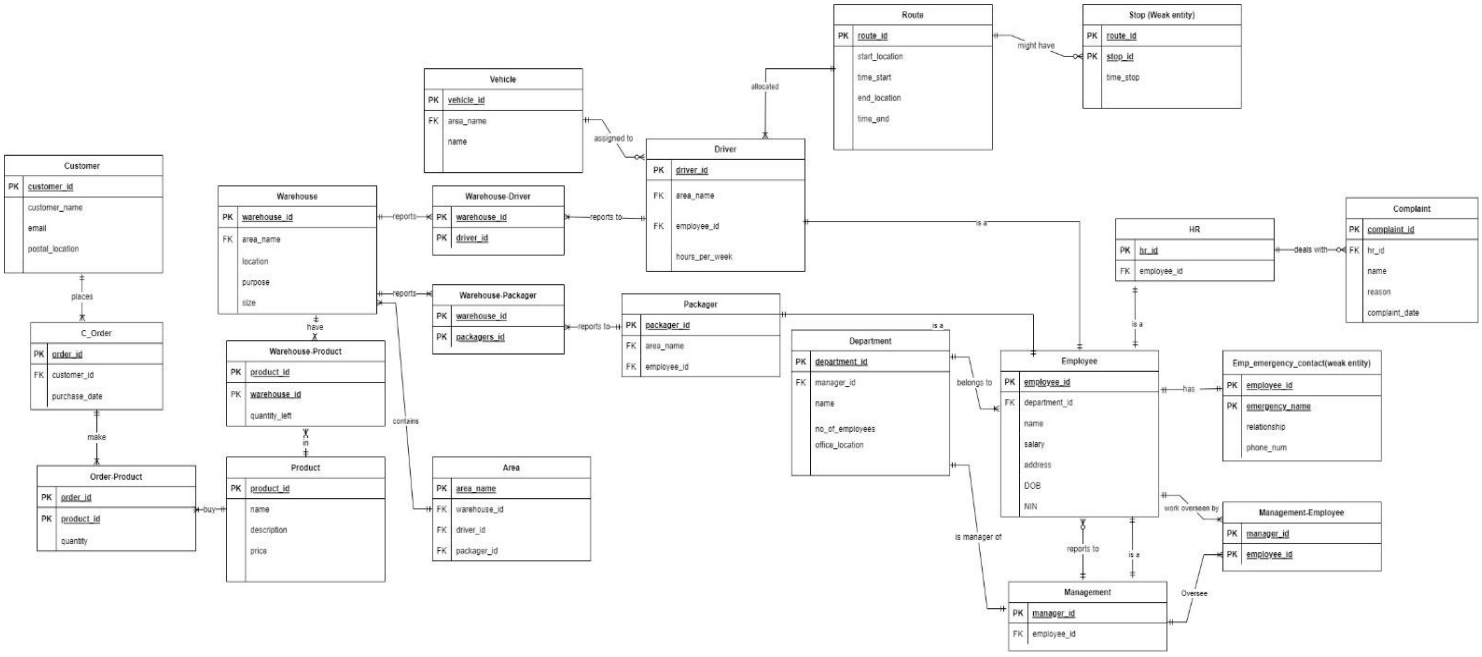
Contents

ERD

UNF:


Route
PK route_id
start_location
time_start
end_location
time_end
might have
Stop (Weak entity)
PK route_id
PK stop_id
time_stop
allocated
Vehicle
PK vehicle_id
FK area_name
name
assigned to
Driver
PK driver_id
FK area_name
FK employee_id
hours_per_week
Customer
PK customer_id
customer_name
email
postal_location
Warehouse
PK warehouse_id
FK area_name
location
purpose
size
reports to
is a
HR
PK hr_id
FK employee_id
deals with
Complaint
PK complaint_id
FK hr_id
name
reason
complaint_date
places
reports to
Packager
PK packager_id
FK area_name
FK employee_id
is a
C_Order
PK order_id
FK customer_id
purchase_date
Department
PK department_id
FK manager_id
name
no_of_employees
office_location
Employee
PK employee_id
FK department_id
name
salary
address
DOB
NIN
has
Emp_emergency_contact(weak entity)
PK employee_id
PK emergency_name
relationship
phone_num
have
contains
belongs to
Product
PK product_id
name
description
price
Area
PK area_name
FK warehouse_id
FK driver_id
FK packager_id
buy
is manager of
oversee work
reports to
is a
Management
PK manager_id
FK employee_id

3NF


Route
PK route_id
start_location
time_start
end_location
time_end
might have
Stop (Weak entity)
PK route_id
PK stop_id
time_stop
allocated
Vehicle
PK vehicle_id
FK area_name
name
assigned to
Driver
PK driver_id
FK area_name
FK employee_id
hours_per_week
Customer
PK customer_id
customer_name
email
postal_location
places
C_Order
PK order_id
FK customer_id
purchase_date
make
Order-Product
PK order_id
PK product_id
quantity
buy
Warehouse
PK warehouse_id
FK area_name
location
purpose
size
reports
Warehouse-Driver
PK warehouse_id
PK driver_id
reports to
reports
Warehouse-Packager
PK warehouse_id
PK packagers_id
reports to
have
Warehouse-Product
PK product_id
FK warehouse_id
quantity_left
contains
in
Product
PK product_id
name
description
price
Area
PK area_name
FK warehouse_id
FK driver_id
FK packager_id
Packager
PK packager_id
FK area_name
FK employee_id
is a
HR
PK hr_id
FK employee_id
deals with
Complaint
PK complaint_id
FK hr_id
name
reason
complaint_date
is a
Department
PK department_id
FK manager_id
name
no_of_employees
office_location
belongs to
Employee
PK employee_id
FK department_id
name
salary
address
DOB
NIN
has
Emp_emergency_contact(weak entity)
PK employee_id
PK emergency_name
relationship
phone_num
work overseen by
Management-Employee
PK manager_id
PK employee_id
is manager of
reports to
is a
Management
PK manager_id
FK employee_id
Oversee

Report:

The ERD shown at top is still not fully normalized, as there are still many-to-many relationships.

The next ERD is already in 3NF as it make it easier to understand the relationships, where I have resolved all the many-to many relationships.

Each employee belongs to only 1 department, while each department can have 1 or more employees. Primary key(PK) is employee_id, which will become the FK in the Drivers, Packagers, HR and Management entities, as this is assigned to each of the employee regardless of their department.This helps for faster query across the departments.

Product and Order have many to many relationship, hence we use bridging entity type OrderedProduct, with composite (order_id, product_id)PK.

Warehouse and Drivers also have many to many relationship because each area can have many drivers and warehouse, hence we use bridging entity type Warehouse-Drivers, with composite (warehouse_id, drivers_id)PK.

Warehouse and Packagers also have many to many relationship, hence we use bridging entity type Warehouse-Packagers, with composite (warehouse_id, packagers_id)PK.

Warehouse and Product also have many to many relationship as each warehouse can have many products, and each product can be contained in many warehouses., hence we use bridging entity type Warehouse-Product, with composite (warehouse_id, product_id)PK.

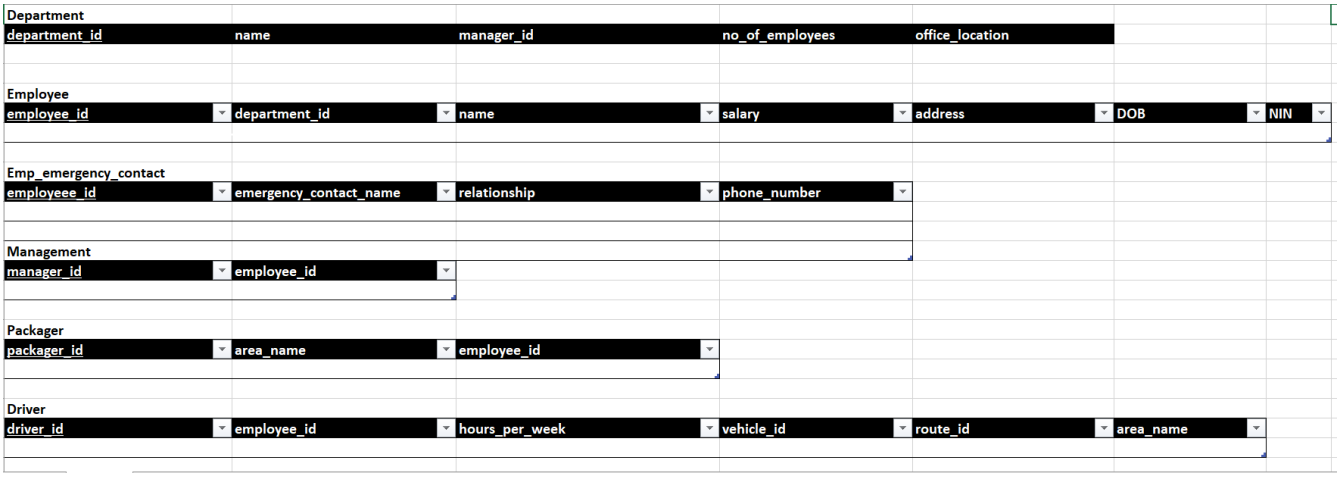
For area, its name is assumed to be unique, thus can be use as primary key.

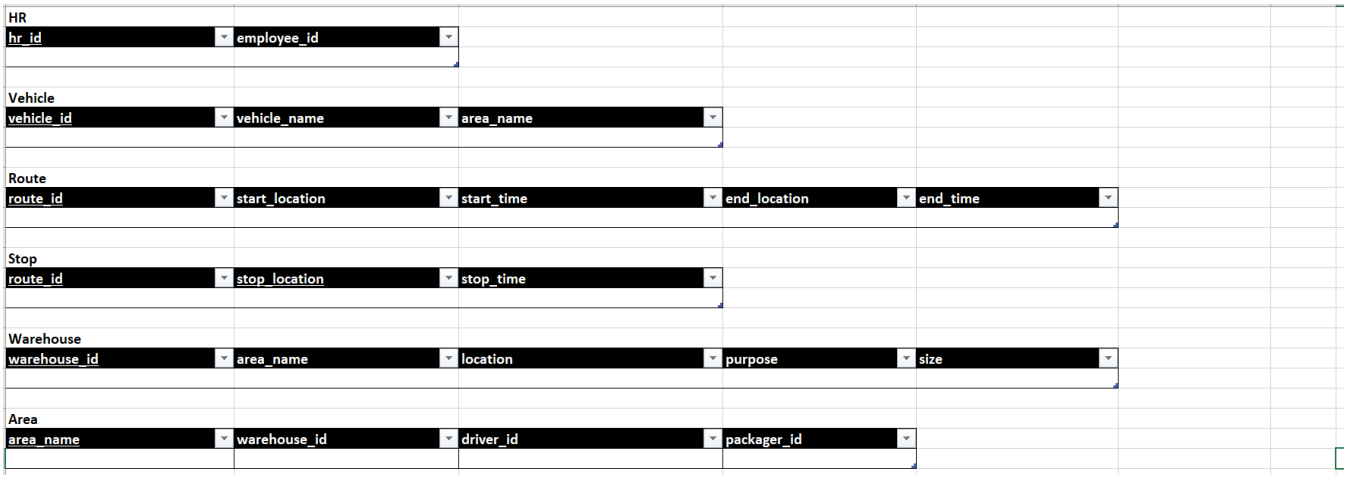
Stops is a weak entity, which can only exist depending on Routes. Routes have full participation, so route_id becomes the FK in Stops. Stops have composite (Routes.route_id, Stops.stop_id) PK.

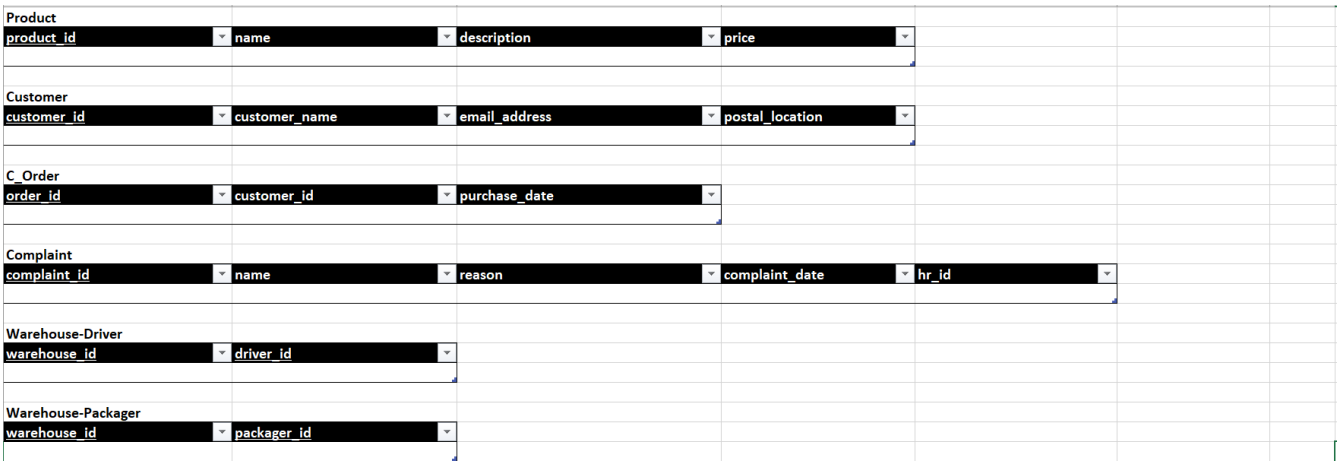
Similarly, Emp_emergency_contact is weak entity which depends on employee. Employee have full participation, so employee_id becomes the FK in Emp_emergency_contact. Emp_emergency_contact have composite (Employee.employee_id, Emp_emergency_contact.emergency_name) PK.

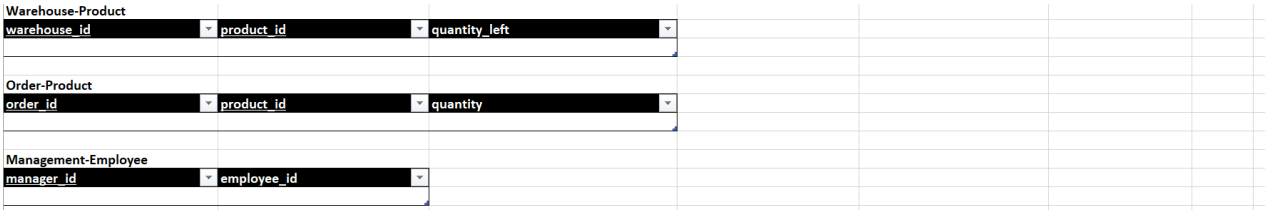
NORMALISATION

The following relations are in 3NF









Report:

UNF- There are several many-to-many relationships including:

- between warehouse_id and driver_id as each warehouse can receive report from 1 or many drivers in the area, and each driver can report to 1 or many warehouse in its area.
- between warehouse_id and packager_id as each warehouse can receive report from 1 or many packager in the area, and each packager can report to 1 or many warehouse in its area.
- between warehouse_id and product_id as each warehouse can contain 1 or many product_id, and each product_id can exist in 1 or many warehouses.
- between product_id and order_id as each product can be purchased from many order as long as the quantity is not 0, and each order can purchase 1 or many products.
- between manager_id and employee_id as each manager can oversee work of 1 or many employees from any departments, and each employee work can be overseen by multipe managers.

1NF- It contains no repeating groups, and value in each column is atomic. To resolve the above mention many to many relationship, bridging entity type is introduce as following:

- Warehouse-Driver
- Warehouse-Packager
- Warehouse-Product
- Order-Product
- Management-Employee

2NF- The relations are already in 1NF. However, there are partial dependencies between attributes to composite PK, including:

- In the Drivers table, the PK is a composite key (driver_id, route_id) and the attributes start_location, start_time, end_location and end_time is only dependent on the route_id part of the PK. This is partial dependencies and needs to be removed by introducing another table, Routes.
- In the Drivers table, the PK is a composite key and the attributes vehicle_name is only dependent on the vehicle_id part of the PK. This is partial dependencies and needs to be removed by introducing another table, Vehicles.
- In Order-Product table, PK is (order_id, product_id). The attributes name, description, price depends only on product_id. Hence this removed to new table, Product, with the product_id being FK in Order-Product table

3NF- The relations are already in 2NF. But, there are transitive functional dependency between non key attributes to other non key attribute, including:

- In the Employee table, the attributes relationship and phone_num depend on the attribute emergency_contact_name, which depends on the PK(employee_id).

  employee_id -> emergency_contact_name

  emergency_contact_name -> relationship, phone_num

  To remove this transitive functional dependencies on PK, attributes of Emergency_contact are move to a new table, Emp_emergency_contact.

- In the Drivers table, the attributes vehicle_name depends on the attribute vehicle_id, which depends on the PK(employee_id).

  employee_id -> vehicle_id

  vehicle_id -> vehicle_name

  To remove this transitive functional dependencies on PK, vehicle_id become the foreign key in the Drivers table, and the attribute vehicle_name is move to a new table, Vehicles.

- In the C_Order table, the attributes customer_name, email_address, postal location depend on the customer_id which depends on the PK(order_id)
  order_id ->customer_id
  customer_id-> customer_name, email_address, postal location
  To remove this transitive functional dependencies on PK, customer_id become the foreign key in the C_order table, and all the attributes dependent on customer_id are move to a new table, Customers.

RELATIONAL SCHEMA

3NF:

	department_id	name	manager_id	no_of_employees	office_location
data type	INT	VARCHAR(30)	INT	INT	VARCHAR(30)
constraint	NOT NULL	NOT NULL	NOT NULL	can be NULL	NOT NULL
default	none	none	none	NULL	none
key	PK AUTO_INCREMENT		FK to Management(manager_id)		

Employee

	employee_id	department_id	name	salary	address	DOB	NIN
data type	INT	VARCHAR(30)	VARCHAR(30)	INT	VARCHAR(30)	DATE	INT
constraint	NOT NULL	NOT NULL	NOT NULL	can be NULL	can be NULL	can be NULL	can be NULL
default	none	none	none	NULL	NULL	NULL	NULL
key	PK AUTO_INCREMENT	FK to Departments(department_id)					

Emp_emergency_contact

	employeee_id	emergency_contact_name	relationship	phone_number
data type	INT	VARCHAR(30)	VARCHAR(30)	INT
constraint	NOT NULL	NOT NULL	can be NULL	can be NULL
default	none	none	NULL	NULL
key	PK, FK to Employees(employee_id)	PK		

Management

	manager_id	employee_id
data type	INT	INT
constraint	NOT NULL	NOT NULL
default	none	none
key	PK AUTO_INCREMENT	FK to Employee(employee_id)

Packager

	packager_id	area_name	employee_id
data type	INT	INT	INT
constraint	NOT NULL	NOT NULL	NOT NULL
default	none	none	none
key	PK AUTO_INCREMENT	FK to Area(area_name)	FK to Employee(employee_id)

Driver

	driver_id	employee_id	hours_per_week	vehicle_id	route_id	area_name
data type	INT	INT	INT	INT	INT	VARCHAR(30)
constraint	NOT NULL	NOT NULL	can be NULL	NOT NULL	NOT NULL	NOT NULL
default	none	none	NULL	none	none	none
key	PK AUTO_INCREMENT	PK, FK to Employees(employee_id)		FK to Vehicles(vehicle_id)	FK to Routes(route_id)	FK to Areas(area_name)

HR

	hr_id	employee_id
data type	INT	INT
constraint	NOT NULL	NOT NULL
default	none	none
key	PK AUTO_INCREMENT	FK to Employee(employee_id)

Vehicle

	vehicle_id	vehicle_name	area_name
data type	INT	VARCHAR(30)	VARCHAR(30)
constraint	NOT NULL	can be NULL	NOT NULL
default	none	NULL	none
key	PK AUTO_INCREMENT		FK to Area(area_name)

Route

	route_id	start_location	start_time	end_location	end_time
data type	INT	VARCHAR(30)	TIME	VARCHAR(30)	TIME
constraint	NOT NULL	can be NULL	can be NULL	can be NULL	can be NULL
default	none	NULL	NULL	NULL	NULL
key	PK AUTO_INCREMENT				

Stop

	route_id	stop_location	stop_time
data type	INT	VARCHAR(30)	TIME
constraint	NOT NULL	NOT NULL	NOT NULL
default	none	none	none
key	PK, FK to Route(route_id)	PK	

Warehouse

	warehouse_id	area_name	location	purpose	size
data type	INT	VARCHAR(30)	VARCHAR(30)	VARCHAR(50)	REAL
constraint	NOT NULL	NOT NULL	NOT NULL	can be NULL	can be NULL
default	none	none	none	NULL	NULL
key	PK AUTO_INCREMENT	FK to Area(area_name)			

Area

	area_name	warehouse_id	driver_id	packager_id
data type	VARCHAR(30)	INT	INT	INT
constraint	NOT NULL	NOT NULL	NOT NULL	NOT NULL
default	none	none	none	none
key	PK	FK to Warehouse(warehouse_id)	FK to Driver(driver_id)	FK to Packager(packager_id)

Product

	product_id	name	description	price	quantity_left
data type	INT	VARCHAR(30)	VARCHAR(50)	REAL	INT
constraint	NOT NULL	NOT NULL	can be NULL	NOT NULL	NOT NULL
default	none	none	NULL	none	none
key	PK AUTO_INCREMENT				

Customer

	customer_id	customer_name	email_address	postal_location
data type	INT	VARCHAR(30)	VARCHAR(50)	VARCHAR(100)
constraint	NOT NULL	NOT NULL	NOT NULL	NOT NULL
default	none	none	none	none
key	PK AUTO_INCREMENT			

C_Order

	order_id	customer_id	purchase_date
data type	INT	INT	DATE
constraint	NOT NULL	NOT NULL	NOT NULL
default	none	none	none
key	PK AUTO_INCREMENT	FK to Customer(customer_id)	

Complaint

	complaint_id	name	reason	complaint_date	hr_id
data type	INT	VARCHAR(30)	VARCHAR(255)	DATE	INT
constraint	NOT NULL	NOT NULL	NOT NULL	NOT NULL	NOT NULL
default	none	none	none	none	none
key	PK AUTO_INCREMENT				FK to HR(hr_id)

Warehouse-Driver

	warehouse_id	driver_id
data type	INT	INT
constraint	NOT NULL	NOT NULL
default	none	none
key	PK, FK to Warehouse(warehouse_id	PK, FK to Driver(driver_id)

Warehouse-Packager

	warehouse_id	packager_id
data type	INT	INT
constraint	NOT NULL	NOT NULL
default	none	none
key	PK, FK to Warehouse(warehouse_id	PK, FK to Packager(packager_id)

Warehouse-Product

	warehouse_id	product_id	quantity_left
data type	INT	INT	INT
constraint	NOT NULL	NOT NULL	NOT NULL
default	none	none	none
key	PK, FK to Warehouse(warehouse_id	PK, FK to Product(product_id)	

Order-Product

	order_id	product_id	quantity
data type	INT	INT	INT
constraint	NOT NULL	NOT NULL	NOT NULL
default	none	none	none
key	PK	PK, FK to Product(product_id)	

Management-Employee

	manager_id	employee_id
data type	INT	INT
constraint	NOT NULL	NOT NULL
default	none	none
key	PK, FK to Management(manager_id)	PK, FK to Employee(employee_id)

3NF:

Department(department_id, name, manager_id, no_of_employees, office_location)

FK manager_id->Management(employee_id)

ON DELETE RESTRICT, ON UPDATE CASCADE

Employee(employee_id, department_id , name,salary, address, DOB, NIN,)

FK department_id->Department(department_id)

ON DELETE RESTRICT, ON UPDATE CASCADE

Emp_emergency_contact(employee_id, emergency_contact_name, relationship, phone_num)

FK employee_id -> Employee(employee_id)

ON DELETE CASCADE, ON UPDATE CASCADE

Management(manager_id, employee_id)

FK employee_id -> Employee(employee_id)

ON DELETE CASCADE, ON UPDATE CASCADE

Packager(packager_id, area_name, employee_id)

FK employee_id-> Employee(employee_id)

ON DELETE CASCADE, ON UPDATE CASCADE

FK area_name -> Area(area_name)

ON DELETE RESTRICT, ON UPDATE CASCADE

Driver(driver_id, employee_id, hours_per_week, vehicle_id, route_id, area_name)

FK employee_id -> Employee(employee_id)

ON DELETE CASCADE, ON UPDATE CASCADE

FK vehicle_id -> Vehicle(vehicle_id)

ON DELETE RESTRICT, ON UPDATE CASCADE

FK route_id -> Route (route_id)

ON DELETE RESTRICT, ON UPDATE CASCADE

FK area_name -> Area (area_name)

ON DELETE RESTRICT, ON UPDATE CASCADE

HR(hr_id, employee_id)

FK employee_id->Employee(employee_id)

ON DELETE CASCADE, ON UPDATE CASCADE

Vehicle(vehicle_id, vehicle_name, area_name)

FK area_name-> Area(area_name)

ON DELETE RESTRICT, ON UPDATE CASCADE

Routes(route_id , start_location, start_time, end_location, end_time)

Stop(route_id, stop_location, stop_time)

FK route_id -> Route(route_id)

ON DELETE CASCADE, ON UPDATE CASCADE

Warehouse(warehouse_id, area_name, location, purpose, size)

FK area_name-> Area(area_name)

ON DELETE RESTRICT, ON UPDATE CASCADE

Area(area_name, warehouse_id, driver_id, packager_id)

FK warehouse_id->Warehouse(warehouse_id)

ON DELETE RESTRICT, ON UPDATE CASCADE

FK driver_id -> Driver(driver_id)

ON DELETE RESTRICT, ON UPDATE CASCADE

FK packager_id-> Packager(packager_id)

ON DELETE RESTRICT, ON UPDATE CASCADE

Product(product_id, name, description, price, quantity_left)

Customer (customer_id, customer_name, email_address, postal_location)

C_Order (order_id, customer_id, purchase_date)

FK customer_id -> C_Order(customer_id)

ON DELETE RESTRICT, ON UPDATE CASCADE

Order-Product(order_id, product_id, quantity)

FK order_id -> C_Order(order_id)

ON DELETE RESTRICT, ON UPDATE CASCADE

FK product_id -> Product(product_id)

ON DELETE RESTRICT, ON UPDATE CASCADE

Complaint(complaint_id, name, reason, complaint_date, hr_id)

FK hr_id->HR(hr_id)

ON DELETE RESTRICT, ON UPDATE CASCADE

Warehouse-Driver(warehouse_id, driver_id)

FK warehouse_id->Warehouse(warehouse_id)

ON DELETE RESTRICT, ON UPDATE CASCADE

FK driver_id->Driver(driver_id)

ON DELETE RESTRICT, ON UPDATE CASCADE

Warehouse-Packager(warehouse_id, packager_id)

FK warehouse_id->Warehouse(warehouse_id)

ON DELETE RESTRICT, ON UPDATE CASCADE

FK packager_id->Packager(packager_id)

ON DELETE RESTRICT, ON UPDATE CASCADE

Warehouse-Product(warehouse_id, product_id, quantity_left)

FK warehouse_id->Warehouse(warehouse_id)

ON DELETE RESTRICT, ON UPDATE CASCADE

FK product_id->Product(product_id)

ON DELETE RESTRICT, ON UPDATE CASCADE

Order-Product(order_id, product_id, quantity)

FK order_id->Order(order_id)

ON DELETE RESTRICT, ON UPDATE CASCADE

FK product_id->Product(product_id)

ON DELETE RESTRICT, ON UPDATE CASCADE

Management-Employee(manager_id, employee_id)

FK manager_id->Management(manager_id)

ON DELETE RESTRICT, ON UPDATE CASCADE

FK employee_id->Employee(employee_id)

ON DELETE CASCADE, ON UPDATE CASCADE

Report:

For most of the PK, I have chosen ID with data type integer, as this can be automatically incremented and thus will be always unique. An exception is for Area, where it is explicitly said that the name is unique although the data type is VARCHAR. For date I have used DATE, and for time stop and time end I used TIME.

For PK and FK, I set the constraint that it cannot be null. Although FK can be set to NULL, but I prefer to set that constraint to NOT NULL, because this will help me to refer to the parent table whenever I need information regarding that FK. This is also important for bridging table such as Warehouse-Product, where the FK form part of the composite PK. So, the default value for the key (NOT NULL) is set to none.

As I work through the normalization process, in 1NF , I removed any repeating group of attributes to prevent update anomalies. For example, each area could have 1 or more warehouse, so instead of querying across all the warehouse field(warehouse1, warehouse2,..), I only need to update it in one location, subsequently the update will occur in all the table.

For weak entity such as Stop, the route_id will never be NULL , but stop_id can be null. Hence we use route_id as FK in Stop.

For each of the FK , I have decided what should be done in case the original data in the parent table is updated or deleted.

For ON UPDATE clause, I have use CASCADE. Example: Updating department_id in table Department would update accordingly in all Employee rows referencing the Department, as department_id is the FK in Employee. Hence this could prevent data inconsistencies.

For ON DELETE clause, I have use RESTRICT. Example: Deleting department_id in table Department would be restricted or rejected if at least one employee links to this department_id. This is essential to prevent data loss. However, for tables that contain employee_id as the FK, I have use ON DELETE CASCADE. If an employee_id is removed from Employee, then any data linking to employee_id should also be deleted. This would make sure that no unnecessary data cluttered the database