Database Systems

Kilburnazon Database Report
COMP23111

Pavel Ghazaryan

University of Manchester
November 11 2022

Created Using LaTeX

Contents

Chapter 1

Entity Relationship Diagram

1.1 Introduction

Entity Relationship Diagrams assist in designing databases through showcasing how entities and their attributes relate to one another. In this section, I will present my design for Kilburnazon delivery company database implementing ERD diagrams. My diagram consists of three main sections: Employee/Workspace, Workload/Route and Warehouseflow/Orders.

1.2 ERD and Report

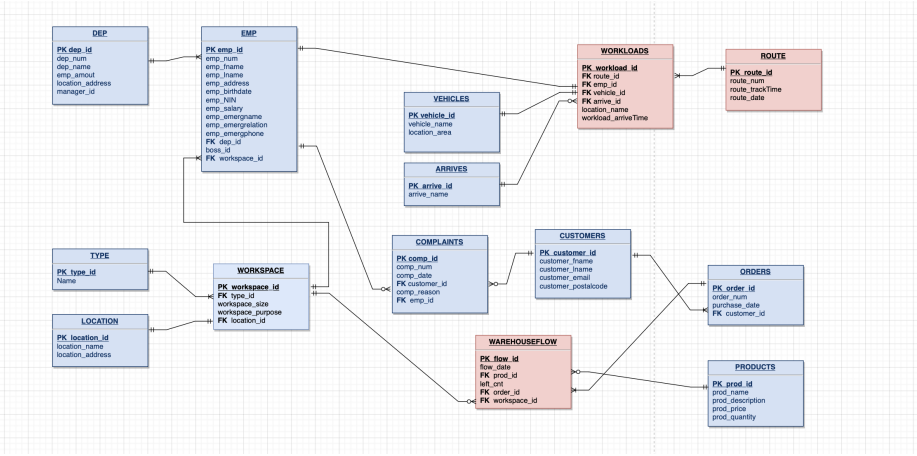

Figure 1.1: The initial look of ERD

Firstly, let's take a look at the Employee/Workspace section. I have made dep-id the primary key for this table as it was not clear whether the department number is unique and unchangeable. I have added all the other necessary attributes and added the department manager as a manager-id, which will contain

id from employee table, meaning that Heads of departments will also be in the employee table. In the Employee table, I have again added a separate emp-id

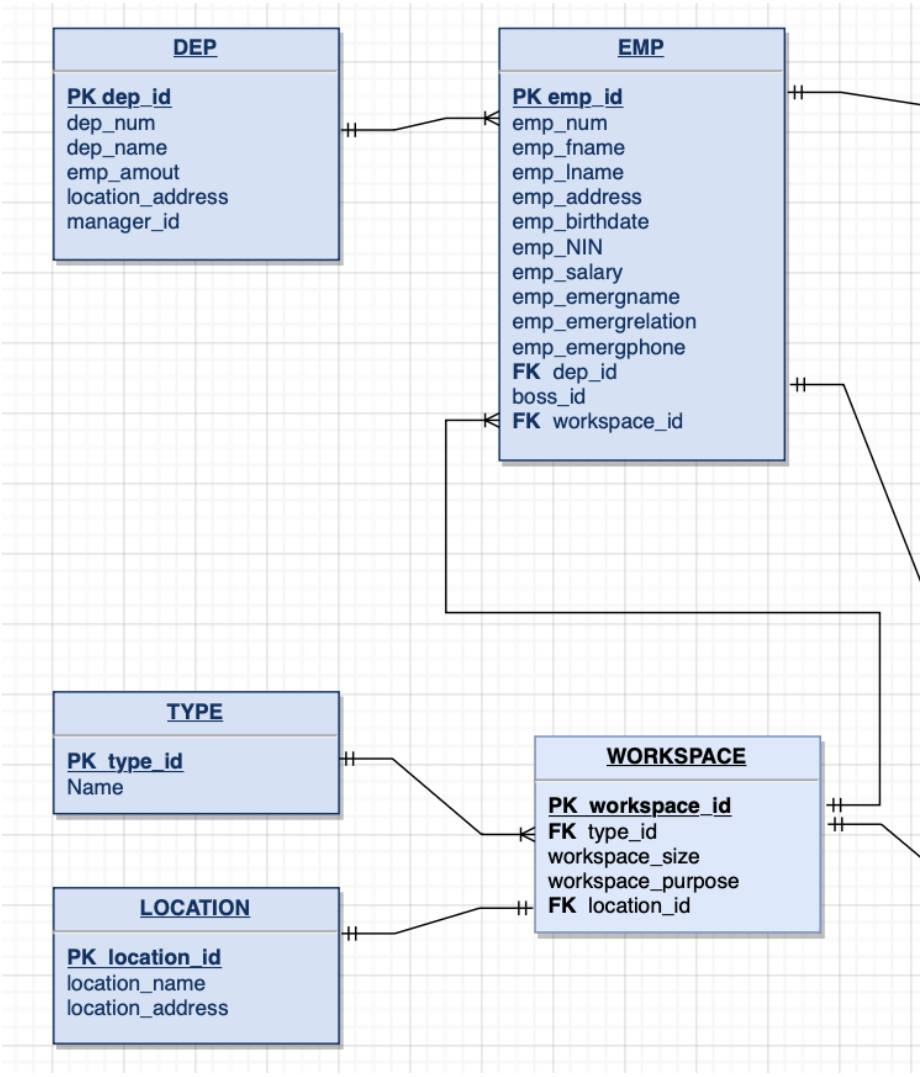

Figure 1.2: Employee/Workspace section

as it is not clear whether the employee number will be unique. One employee may take up the place of another and their employee numbers may change. I have decided to not make a separate table for emergency contact as each employee will have only one emergency contact. I have stored the manager of each employee under the boss id attribute which will contain the employee-id of the manager of the current employee. Consequently, the managers' boss-id will be the emp-id of the Head of the Management department.

I have decided to create a general workspace table and a separate type table which will stand for workspace types: Office or Warehouse. I have also added

a separate location table which will show the name/area and the address of the workspace. I have decided to implement this design as it makes the addition of new Offices and Warehouses in the future more convenient.

In the Workload/Route section, as Kilburnazon is a delivery company, I have

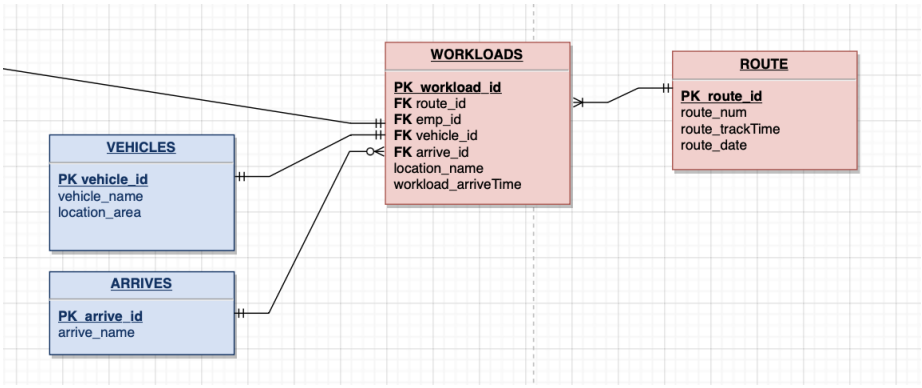

Figure 1.3: Workload/Route section

decided that routes are not pre-determined but are new every time. I have made a general Workloads table which will store the vehicle used, the driver as employee-id, route-id, arrival location and arrival time. I have implemented a separate arrives table which will hold three types of the possible arrival location: start, midpoint and end. Every time a driver arrives at a location, a new row will be created in the Workload table with the same route-id with a different arrive-id and location name and the arrival time will be specified. In the Route table, I have included the trackTime attribute which will show how many hours the route has taken and therefore will present how many hours a driver worked.

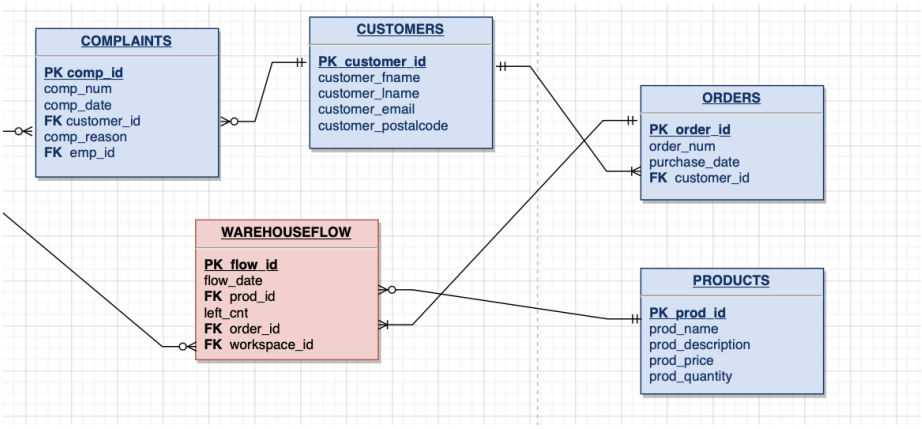

Figure 1.4: Warehouseflow/Orders section

In the Warehouseflow/Orders section, I have implemented a general Warehouseflow table which will be responsible for the flow of the orders. In this table, I specify the product-id and the order-id the product relates to. Warehouseflow

also stores the workspace-id which will show at which warehouse the order is processed and left-cnt will show the quantity of product left at that Warehouse. Customer table will store the necessary customer information. Orders table will show the order purchase date and will store the customer-id of the customer who placed it. Products table stores the necessary information about the product: name, description, price and quantity. I have also included the Complaints table which has all the necessary attributes and foreign keys to store the emp-id who will be dealing with the complaint and the customer-id who filled it.

Word-count: 499

Chapter 2

Normalisation

2.1 Introduction

Normalization is the organization of the database in order to minimize data redundancy in relations and set of relations. In this section, I will analyze my diagram based on 3NF which has the following requirements:

- No repeating groups
- Each value in column is atomic
- No partial dependencies
- There are no transitive dependencies for non-key attributes

2.2 Relations in 3NF and Report

In my diagram it may be seen that all the tables fulfill the first two requirements as based on my design no attribute can contain more than one value and all the groups are unique. We can take a look at the figure below again to be sure. A little bit trickier are the last two points fulfill. Let's analyze the sections one by one.

In the Employee/Workspace section we can see that all the values depend on the primary key of the table. That is, Employee table, every value is dependent on the emp id. When taking a closer look, it may seem that workspace id depends on the department and not the employee d. However, as for Drivers and Packagers there are multiple Warehouse options, the workspace id will also depend on emp id. Which means that the best way to deal with this transitive dependency would be to create a new table which will map workspace id to department id. Therefore, I will be able to keep only the workspace id of the employee in the employee table as the department id can be found based on workspace id. The final look may be seen in the below figure. No partial or transitive dependencies exist at this stage.

Moving forward to the Workspace table, it can be seen that every attribute here depends on the workspace id and there are no partial dependencies, nor

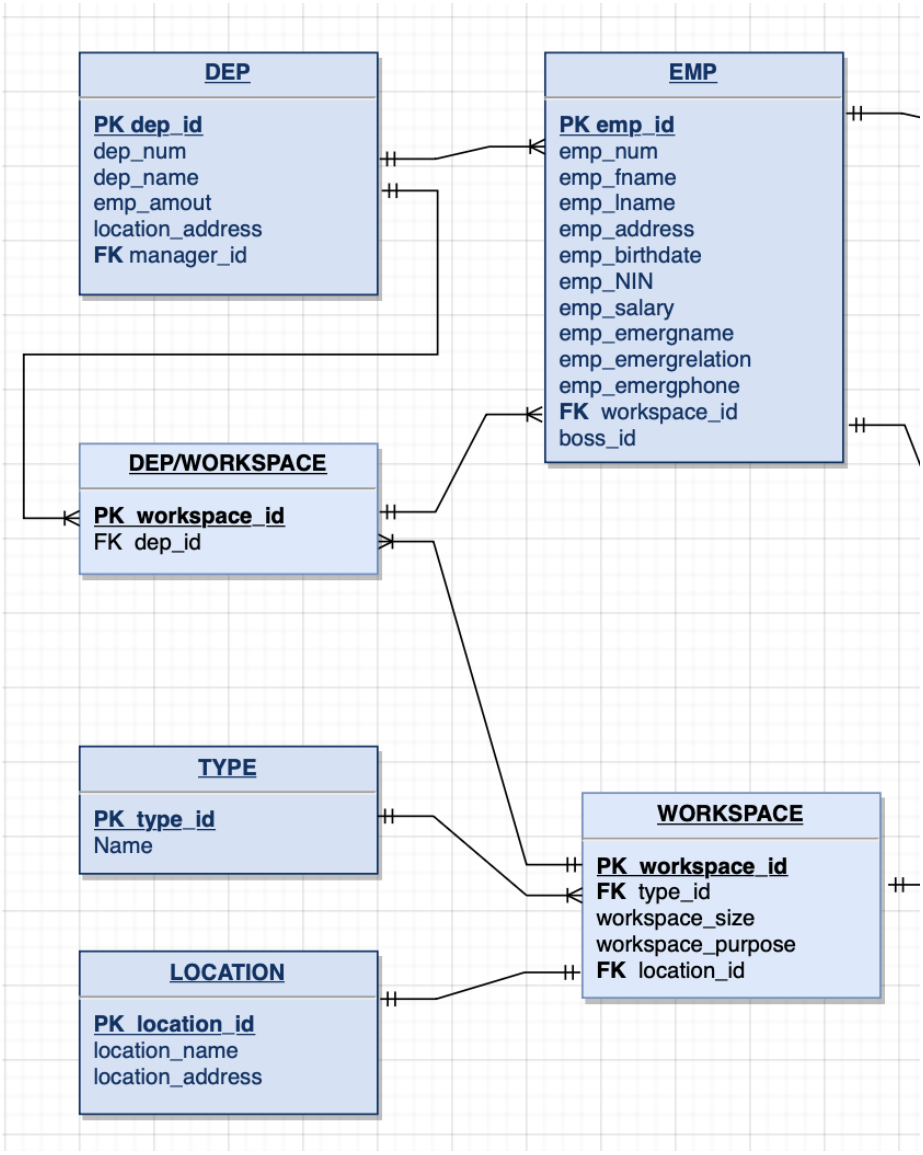

Figure 2.1: Final look Employee/Workspace section

transitive. Every attribute is atomic and there are no redundant groups. The say can be said for Types and Location tables.

In the next section for Workloads/Route, we can see that not everything is dependent on the primary key workload id as we can have the same route with a different arrival type but with different workload id too. Meaning that the route id does not depend on workload id at all. In order to fix this and guarantee uniqueness, we can use a composite primary key with workload id and route id. In this case there will be no partial nor transitive dependencies.

In the Warehousflow/Orders section let's take a look at the Warehousflow table as all the other ones are fairly simple and surely fulfill 3NF requirements.

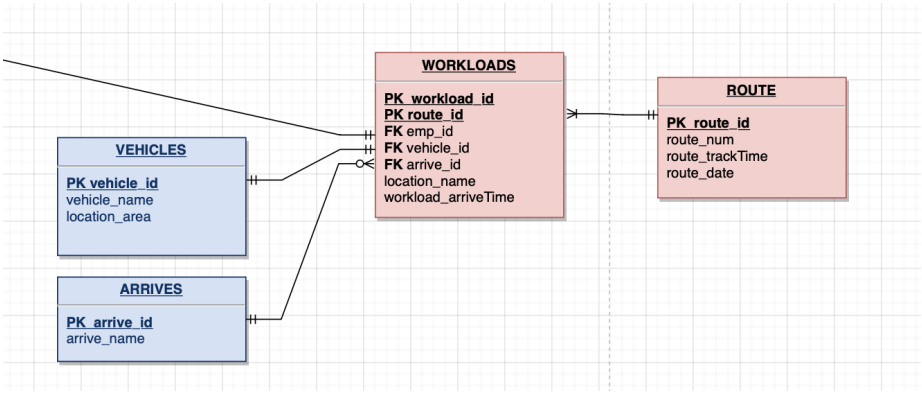

Figure 2.2: Final look Workloads/Route section

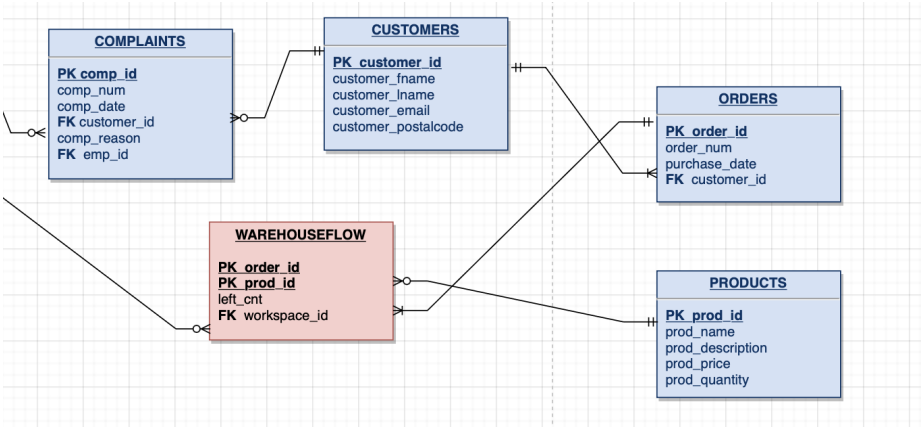

Figure 2.3: Final look Warehousflow/Orders section

In Warehouseflow table it may be seen that not all of the attributes depend on the flow id. For example, order id and product id do not depend at all on flow id. The reason is that this table is a bridging table between orders products so that we will be able to store multiple products within a single order without breaking the atomic cell requirement. This means that in order to remove partial and transitive dependencies we can implement a composite key with order id and product id and remove flow id. I also noticed that flow date has no point as I already store the date of purchase in Orders table so it was a redundant attribute. Complaints table fulfilled all the requirements, so there was no change needed.

The final look:

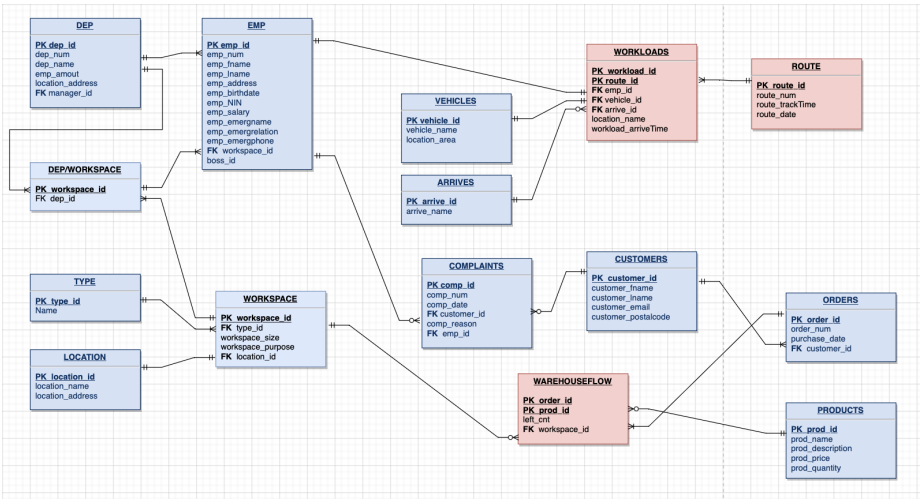

Figure 2.4: Final look of ERD

Word Count: 498

Chapter 3

Relational Schema

3.1 Introduction

A relational schema is a set of relational tables which are related to one another. Through relational schemas the design becomes clearer and the relationships between tables can be viewed. I will use the textual format of relational schemas, in order to represent relationships from my ER diagram.

3.2 Relational Schema and Report

In the section of Department and Employees we have the following relational schemas for each table:

DEP(dep-id, dep-num, dep-name, emp-amout, location-address, emp-id)
FK emp-id $\rightarrow EMP(emp-id)$
$ONDELETECONSTRAINT, ONUPDATECONSTRAINT$

The only foreign key in in the department table is the employee key which is the manager of the department from employee table. None of the attributes can be null at any point.

EMP (emp-id, emp-num, emp-fname, emp-lname, emp-address, emp-birthdate, $\text{emp}_N IN, emp_s alary, emp_e mergname, emp_e mergrelation, emp_e mergphone, workspace-id, boss-id$)
$FKworkspace-id \rightarrow DEP/WORKSPACE(workspace-id)$
FK boss-id $\rightarrow EMP(emp-id)$
ON DELETE CONSTRAINT, ON UPDATE CONSTRAINT

In the employee table there are two foreign keys: first one shows us what is the workspace of the current employee and also connect to the DEP/WORKSPACE table which helps us to find the department of the current employee. Only the boss-id foreign key can be null when the employee is actually a manager of a department who, as provided, have no managers to whom they report.

DEP/WORKSPACE (workspace-id, dep-id)
FK dep-id→ $DEP(dep-id)$
FK workspace-id→ $WORKSPACE(workspace-id)$
ON DELETE CONSTRAINT, ON UPDATE CONSTRAINT

The DEP/WORKSPACE table for matching workspaces with departments has two foreign keys which match the department id with the workplace id. No attributes can be null in this table as we are mapping existing workspaces to existing departments.

WORKSPACE (workspace-id, type-id, workspace$_size, workspace-purpose, location-id$)
$FKtype-id \rightarrow TYPE(type-id)$
FK location-id→ $LOCATION(location-id)$
ON DELETE CONSTRAINT, ON UPDATE CONSTRAINT

In Workspace table there are two foreign keys which show the type of the workspace and the location and name. Workspace size and purpose can be null if the type of Workspace is provided to be an office. The reason being that we are given information only about size and purpose of warehouses.

WORKLOADS (workload-id, route-id, emp-id, vehicle-id, arrive-id, location-name, workload-arriveTime)
FK emp-id→ $EMP(emp-id)$
FK vehicle-id→ $VEHICLES(vehicle-id)$
$FKarrive-id \rightarrow ARRIVES(vehicle-id)$
ON DELETE CONSTRAINT, ON UPDATE CONSTRAINT

In Workloads, table we have three foreign keys which identify the driver (emp-id), the vehicle used and the arrival type (start, midpoint, end). No attribute can be null in this table as they all represent and ongoing workload values.

ORDERS (order-id, order-num, purchase-date, customer-id)
FK customer-id→ $CUSTOMERS(customer-id)$
$ONDELETECONSTRAINT, ONUPDATECONSTRAINT$

Orders table stores only one foreign key which links to the customer who placed the order.

COMPLAINTS (comp-id, comp-num, comp-date, customer-id, comp-reason, emp-id)
FK customer-id → $CUSTOMERS(customer-id)$
FK emp-id → $EMP(emp-id)$
ON DELETE CONSTRAINT, ON UPDATE CONSTRAINT

In the complaints table, I have two foreign keys which show the customer who filled the complain and employee number who will be dealing with the complain.

WAREHOUSEFLOW (order-id,prod-id, left-cnt, workspace-id)
FK workspace-id→ $WORKSPACE(workspace - id)$
$ONDELETECONSTRAINT, ONUPDATECONSTRAINT$

In Warehousflow, there is one foreign key which shows us from which Warehouse is the order being processed. Two primary keys were specified in order to guarantee uniqueness.

Word-count: 489