Laurian Staicu 1696177 IPD9

Step 1: The purpose of the MovinOn database

The database will be capable of sharing data between the three warehouses and any warehouses that the company acquires in the future so that it is easy for the company to share and maintain data, manage employee data, including personal information, salary information, work performance and maintain data about customers who utilize moving and storage services.

The database will manage information about drivers, including their personal information and driving records. The system also needs to store information about the trucks and vans that MovinOn owns and operates.

Step 2: Entities (tables)

1 – employee

2 – drivers

3 – job order

4 – warehouse

5 – vehicle

6 - customer

7- unit rental

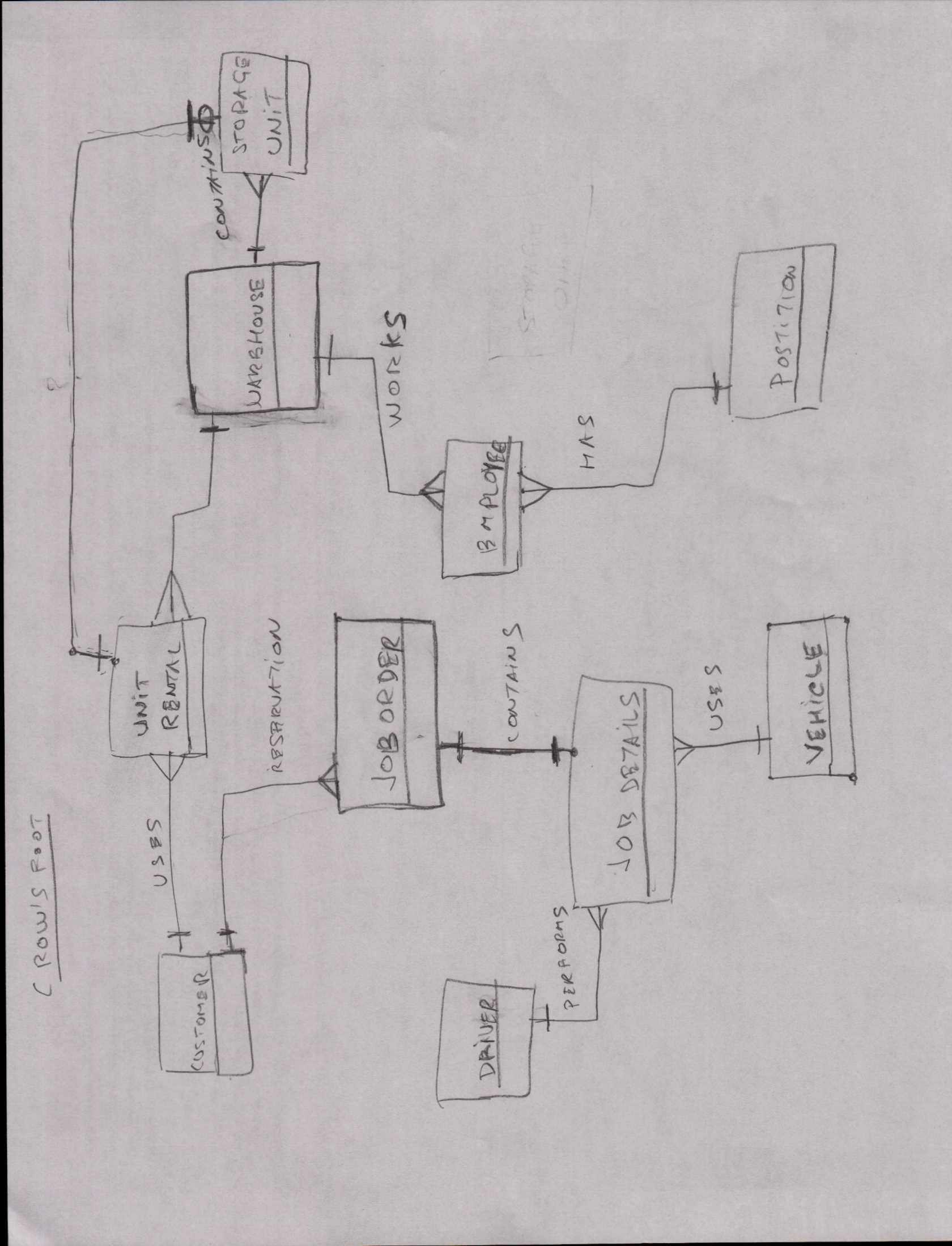
8- job detail

9- storage unit

10 - position

Step 3: attributes (fields) in each entity

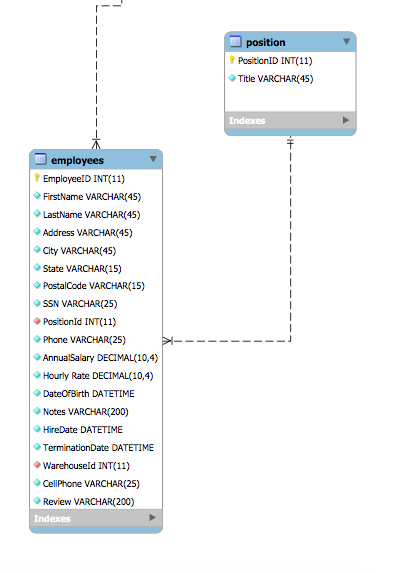
|  |  |  |
| --- | --- | --- |
| 1)employee  1)id  2)first name  3)last name  4)address  5)city  6)state  7)postal code  8)SSN  9)position id  10)phone  11)annual salary  12) hourly rate  13)date of birth  14)notes  15)hire date  16) termination date  17)warehouse id  18)cell phone  19)review | 2) drivers  1) id  2) first name  3) last name  4) address  5) city  6)state  7)postal code  8)SSN  9)phone  10)cell phone  11)rate per mile  12)safety records  13)DOB  14)hire date  15)termination date  16) review  17)notes | 3 – job orders    1) id  2)customer id  3)moving date  4)from address  5) from city  6) from state  7) to address  8) to city  9) to state  10)distance estimated  11) weight estimated  12)packing service  13) heavy packing  14)store items |
| 4 – warehouse    1)id location  2) address  3) city  4) state  5)zip  6)phone  7)climate  8)controlsecurity | 5 – vehicle  1) id  2)license plate  3)number axle  4)color | 6 – customer  1)id  2)company name  3)first name  4)last name  5)address  6)city  7)state  8)postalcode  9)phone  10)balance |
| 7-unit rental  1)unit id  2)customer id  3)warehouse id  4)date in  5)dateout | 8-job detail    1)id  2)vehicle id  3)driver id  4)Mileage  5)WeightActual | 9-storage unit  1)id  2)warehouse id  3)unitsize  4)rent |
| 10-position  1)id  2)title |  |  |



1. design an employee table and any other necessary tables based on this form :

We need to identify the warehouse were the employee works.and also because employee table has to be in normal form we create an entity position Position

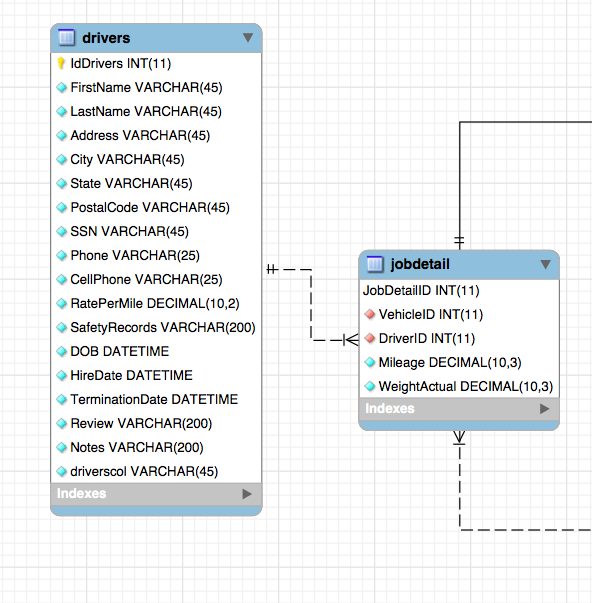
with the attibutes PositonID and Title.



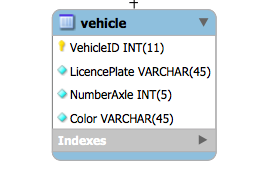
Also because employees can work at any warehouse , we have a dependency constrain , therefore we need a foreign key WarehouseID in employee entity that determines many employees to 1 warehouse relationship.

2. Driver information :

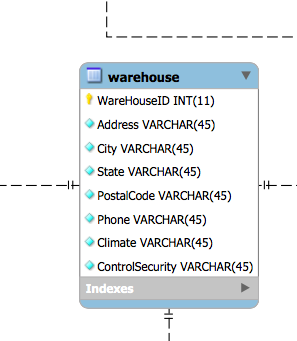
In designing the drivers entity, we have to take in consideration the fact that one drive can be assigned to many job details. Therefore the relationship is one driver to many jobs that implies DriverID as a foreign key in jobdetail entity.



3 Design a table that stores data about the trucks and vans owned by MovinOn. Each vehicle has a unique identification number. The key for vehicle entity is VehicleID.



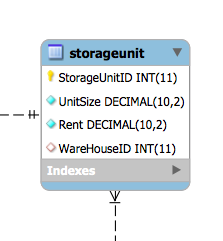
4. Design a table that stores data about warehouses



5. Storage unit entity

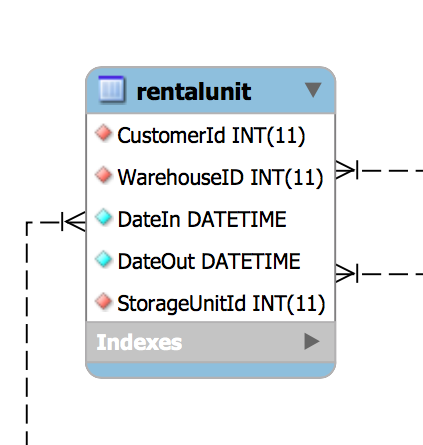
Many storage units belong to warehouse. Therefore we have a dependency constrain in a relationship many to one as many storage unit to one warehouse.

Also we defined the primary key as StorageUnitID.



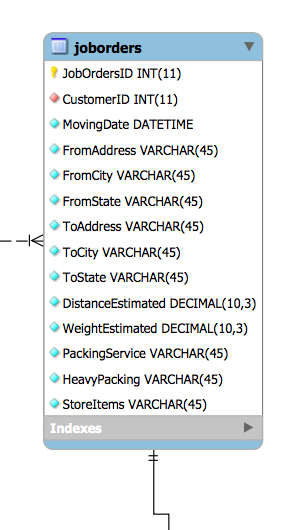
6. Design a rental unit relationship that indicates which unit was rented , what customer , what warehouse, the date the lease started and the date the lease ended.

This relation has to acquire information from 3 entities storage unit, warehouse and customer. To implement it we have to design an entity rental unit that is in a many to one relationship with entities storage unit, warehouse and customer. Consequently, the entity is in third normal form.

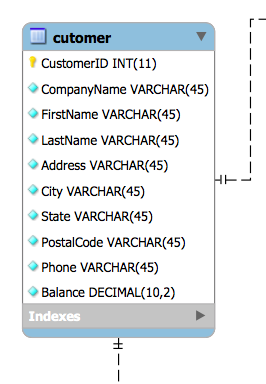


7. Form “job order” is an entity in a one customer to many job orders relationship. One customer can place many job orders and each job order belongs to only one customer.

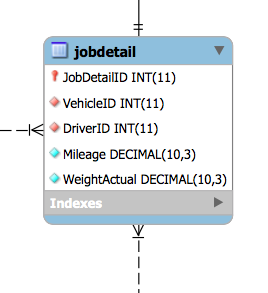
Also we have a one to one relationship with job detail entity, for one job order we have one and only one job detail explained below at number 9.



8. Customers table is a strong entity that participates in two relationships one to many. One customer can place many job orders and one customer can rent can rent many units as explained in rental unit entity.



9. “job detail”. We want to store job detail data separately from job order data. We designed a table that manages the job detail information: vehicle, driver, actual mileage, actual weight and customer. Job detail is a one to one relationship with job orders. Also one driver can be assigned to many job, therefore we have a one to many relationship driver to job details. On vehicle is assigned to many jobs , that results in a relationship one to many with jobs details.



Database MovinON

