Contents

[**1.** **ER diagram.** 3](#_Toc37973332)

[**2.** **Data dictionary.** 4](#_Toc37973333)

[**3.** **Assumptions and Business rules.** 7](#_Toc37973334)

[**4.** **Logical Design** 8](#_Toc37973335)

# **ER diagram.**



# **Data dictionary.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Size / length** | **Value** | | | **Requirement** | **Note** |
|  |  | **Min** | **Max** | **Permitted** |  |  |
| **Entity: AirConditioning** |  |  |  |  |  |  |
| AirConditioningID |  | 1 |  | Integer | Yes | Identifier/Primary key |
| UnitCost |  | Positive number |  | Float | Yes |  |
| BrandName | 255 Characters |  |  | Text | Yes |  |
| Make | 255 Characters |  |  | Text | Yes |  |
| Model | 255 Characters |  |  | Text | Yes |  |
| Weight |  | Positive number |  | Float | No |  |
| Size |  | Positive number |  | Float | No |  |
| Capacity |  | Positive number |  | Float | Yes | It about how powerful conditioner is |
| PowerConsumption |  | Positive number |  | Float | No |  |
| Description | 255 Characters |  |  | Text | No |  |
| AirConType |  | 1 |  | Integer | Yes |  |
| **Entity: Ducted** |  |  |  |  |  |  |
| MaxDistance |  | Positive number |  | Float | Yes |  |
| NumberOfZoneController |  | 1 |  | Integer | Yes |  |
| DuctedWorkSize |  | Positive number |  | Float | Yes |  |
| ReturnAirGrilleSize |  | Positive number |  | Float | Yes |  |
| NumberOfAirDiffuser |  | Positive number |  | Integer | Yes |  |
| **Entity: WallMounted** |  |  |  |  |  |  |
| HP |  | Positive number |  | Float | Yes |  |
| MaxGasTubeLength |  | Positive number |  | Float | Yes |  |
| FoamTubeCoverSize |  | Positive number |  | Float | Yes |  |
| **Entity: WindowMounted** |  |  |  |  |  |  |
| IntakeTubeSize |  | Positive number |  | Float | Yes |  |
| OuttakeTubeSize |  | Positive number |  | Float | Yes |  |
| **Entity: InstallationDetails** |  |  |  |  |  |  |
| TotalCost |  | Positive number |  | Float | Yes |  |
| Quantity |  | 1 |  | Integer | Yes |  |
| **Entity: RequestDetails** |  |  |  |  |  |  |
| RequestDate |  |  |  | Datetime | Yes |  |
| Description | 255 Characters |  |  | Text | No |  |
| **Entity: Location** |  |  |  |  |  |  |
| FirstName | 50 Characters |  |  | Char | Yes |  |
| LastName | 50 Characters |  |  | Char | Yes |  |
| EmailAddress | 100 Characters |  |  | Char | Yes |  |
| PhoneNumberType | 5 Characters |  |  | Char | Yes |  |
| PhoneNumberNumber | 10 Characters |  |  | Char | Yes |  |
| UnitNo | 10 Characters |  |  | Char | Yes |  |
| Street | 50 Characters |  |  | Char | Yes |  |
| State | 50 Characters |  |  | Char | Yes |  |
| PostCode | 50 Characters |  |  | Char | Yes |  |
| Country | 50 Characters |  |  | Char | Yes |  |
| **Entity: Service** |  |  |  |  |  |  |
| ServiceID |  | 1 |  | Integer | Yes | Identifier/Primary key |
| ServiceName | 50 Characters |  |  | Char | Yes |  |
| ServiceType |  | 1 |  | Integer | Yes |  |
| **Entity: Installation** |  |  |  |  |  |  |
| TotalIInstallationCost |  | Positive number |  | Float | Yes |  |
| InstallationDate |  | Not before the requested service date |  | Datetime | Yes |  |
| **Entity: Repair** |  |  |  |  |  |  |
| PosibleRepairDate |  | Not before the requested service date |  | Datetime | Yes |  |
| Problem | 255 Characters |  |  | Text | Yes |  |
| TotalRepairCost |  | Positive number |  | Float | Yes |  |
| **Entity: PerformingServiceDetails** |  |  |  |  |  |  |
| StartingDate |  |  |  | Datetime | Yes |  |
| FinishingDate |  | Not before the installation date or  not before the possible repair date |  | Datetime | Yes |  |
| Role | 150 Characters |  |  | Char | Yes |  |
| **Entity: Payment** |  |  |  |  |  |  |
| PaymentID |  | 1 |  | Integer | Yes | Identifier/Primary key |
| TotalCost |  | Positive number |  | Float | Yes |  |
| Date |  | Not before the requested date |  | Datetime | Yes |  |
| Description | 255 Characters |  |  | Text | No |  |
| **Entity: Employee** |  |  |  |  |  |  |
| EmployeeID |  | 1 |  | Integer | Yes | Identifier/Primary key |
| FirstName | 50 Characters |  |  | Char | Yes |  |
| LastName | 50 Characters |  |  | Char | Yes |  |
| DayOfBirth |  |  |  | Datetime | Yes |  |
| Gender |  |  |  | Bit | Yes |  |
| Skill | 150 Characters |  |  | Char | Yes |  |
| Age |  | 18 | 100 | Integer | Yes |  |
| HireDate |  |  |  | Datetime | Yes |  |
| Street | 50 Characters |  |  | Char | Yes |  |
| State | 50 Characters |  |  | Char | Yes |  |
| PostCode | 50 Characters |  |  | Char | Yes |  |
| Country | 50 Characters |  |  | Char | Yes |  |
| Street | 50 Characters |  |  | Char | Yes |  |
| State | 50 Characters |  |  | Char | Yes |  |
| PostCode | 50 Characters |  |  | Char | Yes |  |
| Country | 50 Characters |  |  | Char | Yes |  |
| **Entity: Invoice** |  |  |  |  |  |  |
| InvoiceID |  | 1 |  | Integer | Yes | Identifier/Primary key |
| DueDate |  | Not before the issue date |  | Datetime | Yes |  |
| IssueDate |  | Not before the issue date |  | Datetime | Yes |  |

# **Assumptions and Business rules.**

|  |  |
| --- | --- |
| Business rules | Assumptions |
| One service has zero or many invoice.  One invoice belongs to one service. | One service might not have any request from customer so there will not have any invoice ,but an invoice has to have what service is. |
| One invoice can be paid by zero or two payments.  One payment pay for only one invoice. | Customer need to wait employee for inspections then pay 40% at front after finish service they will pay the rest. Therefore, in waiting time no payment will be made. |
| One customer has zero or many invoices.  One invoice belong to only one customer. | If the customer request installation service they will not have invoice but invoice need to link to at least one customer. |
| One service has zero or many payments.  One payment belong to only one service. | One service might not have any request from customer so there will not have any payment ,but a payment have to have what it is paying for. |
| One customer makes zero or many payments.  One payment can be made by only one customer. | In the time the customer is waiting for employee come for inspection, the payment amount will not be released to them so they can not make payment. However, payment need to know who make it. |
| One service is undertaken by one or many PerformingServiceDetails.  One PerformingServiceDetails belong to only one service. | Employees can work on installation service and they can work on repair service as well. |
| One PerformingServiceDetails is undertaken by only one employee.  One employee undertake one or many PerformingServiceDetails. |
| One customer makes one or many RequestDetails  One RequestDetails is made by only one customer | When user become a customer so they request at least one service and they can request as many as they can. However, one request has to know who made it. |
| One RequestDetails belong to only one service.  One service has zero or many RequestDetails |
| One service is performed at extra zero or many locations  One location has only one service performing | Extra location for the customer who want installation service at other locations |
| One air conditioning has zero or many installation details.  One installation details belong to one air conditioning | An installation can have many air conditioning and one kind of air conditioning can be installed in many installation services. |
| One installation details belong to only one installation.  One installation contain one or many installation details. |

# **Logical Design**

**Service**(ServiceID, ServiceName, ServiceType)

**Installation(***ServiceID*, TotalInstallationCost, InstallationDate)

foreign key (*ServiceID*) references Service(ServiceID)

**Repair(***ServiceID,* PosibleRepairDate, TotalRepairCost)

foreign key(*ServiceID*) references Service(ServiceID)

**Problem**(Problem, *ServiceID*)

Foreign key(*ServiceID*) references Repair(ServiceID)

**PerformingServiceDetails(***ServiceID, EmployeeID,* StartingDate, FinishingDate, Role)

Foreign key (*ServiceID*) references Service(ServiceID)

Foreign key(*EmployeeID)* references Employee(EmployeeID)

**Employee**( EmployeeID, FirstName, LastName, DayOfBirth, Gender, BasedSalary, Age, HireDate, UnitNo,Steet, State, PostCode, Country, UnitNo,Steet, State, PostCode, Country, EmailAdress)

**PhoneNumber**(Number, Type, *EmnployeeID*)

Foreign key (*EmployeeID*) references Employee(EmployeeID)

**Invoice**(InvoiceID, DueDate, IssueDate, *ServiceID, CustomerID*)

Foreign key (*ServiceID*) references Service(ServiceID)

Foreign key (*CustomerID*) references Customer(CustomerID)

**Payment**(PaymentID, TotalCost, Date, Description, *InvoiceID, CustomerID, ServiceID* )

Foreign key (*InvoiceID*) references Invoice(InvoiceID)

Foreign key (*CustomerID*) references Customer(CustomerID)

Foreign key (*ServiceID*) references Service(ServiceID)

**Customer**( CustomerID, FistName, LastName, UnitNo,Steet, State, PostCode, Country, EmailAdress)

**PhoneNumber**(Number, Type, *CustomerID*)

Foreign key (*CustomerID*) references Customer(CustomerID)

**Location**( *ServiceID,* FirstName, LastName, UnitNo, Steet, State, PostCode, Country)

Foreign key (*ServiceID*) references Service(ServiceID)

**EmailAdress**( EmailAddress, *ServiceID, FisrtName, LastName*)

Foreign key (*ServiceID*) references Location(ServiceID)

Foreign key (*FirstName*) references Location(FirstName)

Foreign key (*LastName*) references Location(LastName)

**PhoneNumber**(Number, Type, *ServiceID, FisrtName, LastName*)

Foreign key (*ServiceID*) references Location(ServiceID)

Foreign key (*FirstName*) references Location(FirstName)

Foreign key (*LastName*) references Location(LastName)

**InstallationDetails**( *AirConditioningID, ServiceID,* TotalCost, Quantity)

Foreign key (*ServiceID*) references Installation(ServiceID)

Foreign key (*AirConditioningID*) references AirConditioning(AirConditioningID)

**RequestDetails**( *ServiceID, CustomerID*, RequestDate, Description)

Foreign key (*ServiceID*) references Service(ServiceID)

Foreign key (*CustomerID*) references Customer(CustomerID)

**AirConditioning**(AirConditioningID, UnitCost, BrandName, Make, Model, Weight, Size, Capacity, PowerConsumption, Description, AirConType)

**Ducted**( *AirConditioningID*, MaxDistance, NumberOfZoneContrller, DuctedWOrkSize, RetunAirGrilleSize, NumberOfAirDiffuser)

Foreign key (*AirConditioningID*) references AirConditioning (AirConditioningID)

**WallMounted**( *AirConditioningID,* HP, MaxGasTubeLength, FoamTubeCoverSize)

Foreign key (*AirConditioningID*) references AirConditioning (AirConditioningID)

**WindowMounted**( *AirConditioningID*, IntakeTubeSize, OuttakeTubeSize)

Foreign key (*AirConditioningID*) references AirConditioning (AirConditioningID)

# **Part A. Database implementation using MS Access.**

## **Normalization**

* 1. **Map the ERD.**

**CUSTOMER**(CustomerID, CustomerName, Street, Address, City, PostCode, CustomerContactNum)

**JOB\_REQUEST**(JobID, JobRequestDate, DateScheduled, JobStartDate, JobEndDate, JobType, TotalCharges, *CustomerID*)

Foreign key(*CustomerID*) references CUSTOMER(CustomerID)

**EMP\_JOB**(*JobID*, *EmployeeID*, EmpStartDate, EmpFisnishDate)

Foreign key(*JobID*) references JOB\_REQUEST(JobID)

Foreign key(*EmployeeID*) references EMPLOYEE(EmployeeID)

**EMPLOYEE**(EmployeeID, EmployeeName, EmployeeStartDate)

**INVOICE**(*InvoiceID*, InvoiceDate, InvoiceAmount)

Foreign key(*InvoiceID*) references JOB\_REQUEST(JobID)

**PAYMENT**(PaymentID, PaymentDate, PaidAmount, *InvoiceID*)

Foreign key(*InvoiceID*) references INVOICE(InvoiceID)

**INSTALLATION**(*JobID*, HP, Address, InstallationType, *AirID*)

Foreign key(*AirID*) references AC\_UNIT (AirID)

**AC\_UNIT**(AirID, Make, Model, HP)

**REPAIR**(*JobID*, Problem, *InstallationID*)

Foreign key(*JobID*) references JOB\_REQUEST(JobID)

Foreign key(*InstallationID*) references INSTALLATION(JobID)

**WALL\_WINDOW\_MOUNTED**(*JobID*, Location)

Foreign key(*JobID*) references INSTALLATION(JobID)

**DUCTED**(*JobID*)

Foreign key(*JobID*) references INSTALLATION(JobID)

**ZONE**(ZoneName, *JobID,* NumOutlets)

Foreign key(*JobID*) references DUCTED(JobID)

* 1. **Functional dependencies and normalization.**

***Relation***: **EMPLOYEE**(EmployeeID, EmployeeName, EmployeeStartDate)

EmployeeID is a primary key that is an identifier of each employee. Every EmployeeID has only one EmployeeName and has only one EmployeeStartDate when this person was started working. Therefore, the duplex group will not exist in this relation. The conditions of being 1NF are that:

* + Containing a primary key
  + No duplex groups

This relation has a primary key and no duplex group; therefore, this relation is 1NF.

EmployeeID is a primary key while other attributes can not make sure it is unique. The reason for that, EmployeeName can be duplexed because people can have the same name and deferent people can have the same EmployeeStartDate. Therefore, there is only a single candidate key which is EmployeeID. The conditions of being 2NF are that:

* + The relation is 1NF
  + It does not have any partial dependencies.

This relation is 1FN and it has only a single candidate key; therefore, this relation is 2NF.

EmployeeName can not determine any other attributes because employees can have the same EmployeeName and similar with EmployeeStartDate. As a result, there will not have transitive dependencies. The condition of being 3NF is that:

* + The relation is 2NF
  + There are no transitive dependencies.

This relation is 2NF and it has no transitive dependencies; therefore, this relation is 3NF.

***Relation***: **AC\_UNIT**(AirID, Make, Model, HP)

AirID is a primary key that is an identifier of each AC\_UNIT. Every AC\_UNIT is made by only one maker and has only one Model and HP. Therefore, the duplex group will not exist in this relation. The conditions of being 1NF are that:

* + Containing a primary key
  + No duplex groups

This relation has a primary key and no duplex group; therefore, this relation is 1NF.

AirID is a primary key while other attributes can not make sure it is unique. The reason for that, Make can be duplexed because many AC\_UNITs can have the same maker, Model or HP. Therefore, there is only a single candidate key which is AirID. The conditions of being 2NF are that:

* + The relation is 1NF
  + It does not have any partial dependencies.

This relation is 1FN and it has only a single candidate key; therefore, this relation is 2NF.

Make can not determine any other attributes because AC\_UNIT can have the same Make and similar with Model and HP. As a result, there will not have transitive dependencies. The condition of being 3NF is that:

* + The relation is 2NF
  + There are no transitive dependencies.

This relation is 2NF and it has no transitive dependencies; therefore, this relation is 3NF.

## **Relational database implementation**

### **Data integrity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Table: JOB\_REQUEST** | | | |
| **Attribute** | **Data type** | **Integrity constraint implemented** | **Error message** |
| JobID | long | Primary key and AutoIncrement |  |
| JobRequestDate | Datetime | Not null | You must enter a value in JOB\_REQUEST.JobRequestDate field |
| DateScheduled | Datetime | Not null  *This Integrity constraint is at Tables level:*  [DateScheduled]>=[JobRequestDate]  And [JobStartDate]>=[DateScheduled]  And  [JobEndDate]>=[JobStartDate] | You must enter a value in JOB\_REQUEST.DateScheduled field  *Another message:*  Make sure end date not after start date, start date after scheduled date, scheduled date before request date |
| JobStartDate | Datetime | You must enter a value in JOB\_REQUEST. JobStartDate field  *Another message:*  Make sure end date not after start date, start date after scheduled date, scheduled date before request date |
| JobEndDate | Datetime | You must enter a value in JOB\_REQUEST. JobEndDate field  *Another message:*  Make sure end date not after start date, start date after scheduled date, scheduled date before request date |