# Office Wizard Data Warehouse Design

(INF3007 – Assignment 3 – Section 1)

## Introduction

Office Wizard is a local company which specialises in supplying office equipment and stationeries to corporate customers. After years of managing the records manually, the company decides to computerise the database. In addition to the main operational database for daily use, Office Wizard also desires to have a data warehouse with the intention of storing archival or obsolete data after it is no longer needed.

This report intends to explain a proposed design for the data warehouse schema in order to provide a consolidated enterprise view of the inhabiting data. In addition, the warehouse also aims to satisfy the information needs of key decision makers within the business. The report will explain subject areas covered by the proposed schema in order to provide indication how these subjects satisfy the information needs. Alongside the theoretical information, an ER diagram and dimensional model matrix of the schema will also be embedded in order to illustrate the design and convey how different entities of data will be related.

## Business Dimensional Lifecycle Basics

While there are many possible methodologies for database warehouse design, the one chosen for this particular scenario was the ‘Business Dimensional Lifecycle’ pioneered by Ralph Kimball. This style of design is also known as ‘Dimensional Lifecycle’ or ‘Dimensional Modelling’. While this report is not centred on theory or research, it is still important to know the basic components of Dimensional Modelling in order to gain a further understanding of how this schema was developed. For this section, the Textbook ‘Database Systems: A Practical Approach to Design, Implementation, and Management (6th Edition)’ will be referenced

The guiding focus associated with the Dimensional Lifecycle relates to meeting the information requirements of the enterprise. This is achieved by building a single, integrated, user-friendly, high-performance infrastructure. The information within this infrastructure is then delivered in specified increments according to the user's desire. The ultimate goal of this methodology does not stop at data warehouses themselves. It is also important for the lifecycle to deliver query tools, report applications and advanced analytics as well as the necessary training and support for the relevant users. While 'Dimensional Modelling' may be a common nickname used for the methodology, it is important to clarify that in the literal sense, these names only refer to a single stage of the lifecycle. Knowing these stages and their order is not important but the fact remains that 'Dimensional Modelling' only refers to the design of the data warehouse schema and not the lifecycle methodology as a whole.

To start with a basic definition, Dimensional Modelling refers to a logical design technique aiming to present the data in a standard, intuitive form to enable high performance access. Generally speaking, the dimensional model is similar to an ER diagram and variations thereof. Regardless of what notation is used, both ER and dimensional models depict tables and how they might be related to one another. In this case, a Dimensional Model consists of at least one large fact table as well as set of smaller dimension tables. Typically, dimension tables contain a non-composite Primary Key with no Foreign Keys while each fact table holds Foreign Keys from different dimensions. This tends to structure the tables in a 'star-like' manner. This structure is fittingly called a 'Star Schema' because the fact tables often appear in the centre while they are surrounded by their respective dimensions.

## Preparation

Previously, a design for the main operational database was proposed for Office Wizard. Of course, this design included an EER diagram which depicts the possible entities and how they are related. While this EER has taken several forms through the design process with different characteristics, the most accepted version of the diagram has been chosen as a basis for the Data Warehouse. After an iteration was chosen, a revision was made to depict only the entities which might provide some use when designing the warehouse schema. In the process of this revision, a discussion was held as to which tables should be removed or kept. For what tables that were kept, it was also considered whether the table should be classified as a dimension or fact. However, this discussion took place before the design itself and it was made clear that the classifications were up for interpretation. In between the discussion and the design, a group E-Mail was sent in order to summarize the conclusions as to which tables should be depicted as fact or dimension. When describing each table and justifying its classification, the information from that E-Mail will be incorporated into the report.

The accepted operational EER diagram and the revised schema diagram have been embedded as Appendices 1 and 2 respectively.

## Chosen Tables

This section is going to provide a description of each table depicted in the schema. This description will include reasons as to why it was chosen and how it was classified as well as an example of a possible query that the table could assist in answering. While the overall content will be similar, the Dimension and Fact tables will be explained separately.

First, it should be forewarned that in comparison to the draft EER, certain liberties were taken to comply with the basic definition of a star schema. Some attributes have also been rephrased to clarify their meaning. Some EER tables may have also been merged where practical in order to show regard for the de-normalization encouraged by dimensional modelling. Regardless, this schema still opts to remain faithful to the source EER as much as possible. In other words, no unnecessary tables or attributes have been depicted which are not already present. On the downside, any pre-existing redundancy or anomalies will have also likely carried over to the schema with no measures taken to fix unless necessary for the schema.

Before the table descriptions, a summary table will be embedded on the next page in order to list which entities are classified as fact and dimension.

### Summary Table

|  |  |
| --- | --- |
| Dimension | Fact |
| Product | SupplierOrderItem |
| Supplier | SupplierOrderPayment |
| SupplierOrder | Allowance |
| TaxBracket | Payslip |
| Payment | CustomerOrderPayment |
| Category | PaymentTime |
| Manufacturer | CustomerOrderItem |
| Customer | ProductDetails |
| CustomerOrder | CustomerOrderDetails |
| Employee | Shift |
| AllowanceType | EmployeeDOB |
| Position | PositionAssignment |
| Timestamp | SupplierOrderDetails |
| Quotation | QuotationDetails |
|  | QuotationItem |
|  | UserAccount |

### Dimension Tables

#### Product

The Product is the database entity referring to any sort of product being sold by Office Wizard whether in the store or made available to order. As far as the containing attributes go, the Warehouse depiction of a Product is the same as in the operational ER. A Product is identified using an ID number. The entity also stores the product name, a general description, a quantity description (eg. Ream of paper), the default unit price, the item status, how many units are currently available, the suggested number of units to re-order at, and the maximum percentage this individual item can be discounted at.

The reason why a Product was considered a dimension is that while it does not represent an important business process in itself, it is still part of buying and selling which are perhaps the most important processes of all. This entity stores information about the product itself while its participation in processes is stored in the fact tables.

The 'Product' dimension is associated with the 'SupplierOrderItem', 'CustomerOrderItem', 'ProductDetails' and 'QuotationItem' fact tables

#### Supplier

The Supplier is the database entity referring to a company responsible for supplying or wholesaling various products to Office Wizard so that they can be sold to customers in the store. The ‘Warehouse’ Supplier remains unchanged from the ‘Operational’ Supplier. It is identified with an ID number. In addition, the entity stores names for both the supplying company itself and the contact person speaking on behalf of that company. A Supplier also has a physical address as well as phone and fax numbers.

The reason why a Supplier was considered a dimension is that while retaining their contact information is important, they do not directly represent an important business process within Office Wizard. They only represent a single part of various processes such as requesting quotations and ordering goods.

The ‘Supplier’ dimension is associated with the 'SupplierOrderDetails' and 'QuotationDetails' fact tables

#### SupplierOrder

The Supplier Order entity is used to represent the basic instance of an order made to a given supplier in order to obtain stock to sell to customers. While a Supplier Order is identified by a composite ID including a sequential number per Supplier, the composition was merged into a single ID for the warehouse. This was both for simplicity and to comply with the dimensional model's rules in that Dimensions cannot have composite keys. While all of the original attributes are depicted in the model as a whole, they are distributed into the 'SupplierOrderDetails' facts table. The dimension table itself only includes an ID and description while the details fact includes the supplier fulfilling the order, the employee who made the order, the quotation which led to the order, and the dates of order and delivery. The payment amount and verification is held separately in the 'SupplierOrderPayment' fact table.

While the name of this entity would suggest a fact table, it was still chosen as a dimension. This can be interpreted as the level of grain involved but the more practical reason was to maintain compliance with the Star schema while still enabling the same amount of information to be stored and associated. While the order made to a Supplier is indeed an important business process, the main information regarding this process is held in the fact tables. This dimension only serves to indicate that the supplier order happened in the first place. By itself, we do not need to know the details regarding this event. If the details are required, they are stored in the mentioned fact tables.

The Supplier Order is not the only process that is depicted in this way. Customer Order and Quotation are important examples with Product and Payment being secondary examples. The 'SupplierOrder' dimension is associated with the 'SupplierOrderDetails', 'SupplierOrderItem'  
and 'SupplierOrderPayment' fact tables.

#### TaxBracket

While some people may question the storage of tax brackets in an operational database, it may be necessary for a system to incorporate these in some fashion in order to automatically calculate tax. As the tax brackets are often subtly adjusted per financial year, it is justifiable to store archive data of tax brackets in past financial years. As it was a fairly simple entity in the database, all of the attributes are retained. However, similar to 'SupplierOrder', the Tax Bracket had a composite key which was merged for its use as a dimension. Originally, Tax Brackets were identified by a sequential number per year. Now a single ID is used with the year and bracket numbers only being informative. As expected, the Tax Bracket also stores the lower and upper income limits as well as the percentage deducted from payments that fall under this bracket.

The reason why the Tax Bracket was depicted as a dimension is because they only serve as variables in calculations and are not a central component of any processes. A payslip can still occur without tax but sadly, this does not happen often.

The 'TaxBracket' dimension is associated with the 'Payslip' and 'Allowance' fact tables as these both relate to income earned by the Employee.

#### Payment

This is not to be confused with 'Payslip'. Table sorting aside, the difference between the two is that while a 'Payslip' is an example of a process involving payments, the 'Payment' dimension table represents the payment itself from the perspective of financial institutions. The 'Payment' entity in the Database was quite simple being identified by a reference number while also storing the timestamp and amount. As the key was non-composite to begin with, this remains unchanged. The only difference is that the payment timestamp is now offloaded to a separate fact table. This is due to the fact that it is common practice to include Timestamps as a dimension. Therefore, the timestamp of a payment must be stored separately from its amount.

A 'Payment' is considered a dimension in spite of it being an important process in general. However, the reason it is not a fact is because a payment can be apart of several different business processes in different contexts. Accounting for those contexts separately would introduce more redundancy and storage than what is necessary regardless if the schema is accounted for. Therefore, a Payment in itself is considered a dimension. The context depends on which fact table it appears in.

The 'Payment' dimension is associated with the 'SupplierOrderPayment', 'Allowance', 'Payslip', 'CustomerOrderPayment' and 'PaymentTime' fact tables

#### Category and Manufacturer

Among the group, the inclusion or even mere existence of these entities has been subject to disagreement. While the design can be subject to interpretation, the fact remains that storing them as loose attributes rather than full entities can lead to anomalies if not managed correctly. Regardless of the operational database, these 'entities' are being depicted as dimensional tables not only to prevent potential anomalies but also that they are associated with processes and queries related to identifying and sorting products.

The reason these two dimensions are being described together is because while they represent two different things, they are still associated with the same processes and have the same set of attributes. Both dimensions are identified with a single ID number while also storing a name and description.

The 'Category' and 'Manufacturer' dimensions are only associated with the 'ProductDetails' fact table with the purpose of sorting a given Product according to what category it is part of and who it is manufactured by.

#### Customer

The Customer dimension represents a person or organization which purchases goods from  
Office Wizard. Be reminded that this does not necessitate every single customer will be stored on file. This entity is used to store those customers who must be stored on file for whatever reason necessary. An example of this is ordering goods for delivery. While all of the entity attributes have been carried over to the dimension, the subclassing concept has been removed. This means that the 'Customer' superclass was merged with the 'CustomerCompany' subclass. The Customer dimension is identified with a single ID while storing the name, gender, E-Mail address, delivery address, phone number and fax number. As the Superclass and Subclass has been merged, the dimension also stores the name of the contact person and the amount of credit they have.

The reason a Customer is depicted as a dimension is similar to that of the Supplier. While they are a major contribution to important business processes, a Customer does not directly represent a process in itself. Sometimes they may not even be represented with buying goods at all if they are not on file. While they are a necessary component of the goods purchasing process, they are still only one part of the process and do not represent the whole.

The 'Customer' dimension is only associated with the 'CustomerOrderDetails' fact table

#### CustomerOrder

Where Office Wizard takes out Supplier Orders in order to acquire their stock, the Customer Order is the inversion of this. The 'CustomerOrder' entity represents the process in which a customer buys goods. This can vary from simply purchasing items in the store or having them delivered to a given location. As it were in the original ER, a Customer Order is identified with a single ID unlike the Supplier Order where the ID is Supplier-specific. This difference is because oddly enough, having the customer on file is not really necessary for the transaction to take place. Most of these sales will involve individuals going into the store, buying items and then going about the rest of their day. Most of these individual customers will not want their personal information retained.

Between the operational entity and warehouse dimension, all of the non-key attributes are retained. However, the 'DeliveryOrder' subclass has been merged into the superclass for this dimension. In addition to the ID, a Customer Order also has a base total cost, amount subtracted due to discounts, the net total cost, the order status, sale mode and order type. As a result of the subclass merge, the dimension also contains attributes for the delivery address and the amount charged for the service.

For reasons as to why 'CustomerOrder' is a dimension table instead of a fact table,  
refer to 'SupplierOrder'.

The 'CustomerOrder' dimension is associated with the 'CustomerOrderDetails',  
'CustomerOrderItem' and 'CustomerOrderPayment' fact tables.

#### Employee

The 'Employee' entity represents any person which has some level of professional association with the Office Wizard organization. An Employee can take on many different roles including sales, management, IT and transport. What is common between the roles is that all Employees are stored in the Database. The attributes stored are mostly unchanged between the operational entity and the dimension. An Employee is identified using a single ID while also storing their name, gender, address, home phone and mobile phone. The Employee's username and password for their system account are not stored with the main entity unlike the operational ER. Instead, they are diverted to a  
User Account fact table so that past username/password combinations for a given Employee can be archived. The Employee's Date of Birth is handled under similar circumstances where it is split to a fact table which holds the DOB of Employees.

The reason that the Employee is depicted as a dimension table is because that while Employees of all sorts partake and oversee a number of important processes, they still do not represent an important process in themselves. For example, an Employee works a shift, receives a payslip, requests a quotation. In addition, Employees of Office Wizard oversee both Customer and Supplier orders.

The 'Employee' dimension is among those associated with the most fact tables. As such, they are associated with the 'Allowance', 'Payslip', 'CustomerOrderDetails', 'Shift', 'EmployeeDOB',  
'PositionAssignment', 'SupplierOrderDetails', 'QuotationDetails' and 'UserAccount'

#### AllowanceType

In addition to their normal monthly pay, an Employee may sometimes receive bonuses for various reasons. These reasons include special occasions, sick leave or simply for good performance. The 'AllowanceType' entity is used to store the various different types of allowances that can be granted to an Employee. The attributes stored have remained unchanged between operation and dimension. A type of allowance is identified with a single ID number along with a unique name to identify on face value. An allowance type also includes an internal description and a category as to how frequently it can occur. Frequencies include weekly, monthly, quarterly and yearly.

The reason that the Allowance Type is classified as a dimension is because while they are crucial in the granting of allowances, the entity as a whole is only used to store the types of allowances available. When a manager for example grants an allowance, they must choose among the different types stored in the table and reference them accordingly. Therefore, the allowance type is only a single component of the allowance process. It does not dictate the exact time and amount of the grant nor which Employees are eligible.

The 'AllowanceType' dimension is predictably associated with the 'Allowance' fact table. It is only associated with that particular fact table and no others.

#### Position

When an Employee works at Office Wizard, they are typically assigned into the role they best fit. It is also possible that an Employee will work in several different positions during their career such as after a promotion. Similar to how Allowance Types are invoked in Allowance grants, the Position stores the data of different individual job roles than can be assigned to Employees. The 'Position' dimension has the same attributes as the operational entity counterpart. A Position is identified the same way as an allowance type. This consists of a single numeric ID for the Primary Key with the name functioning as an Alternative Key. The Position also stores the hourly rate an Employee of that position is paid, as well as the desired sales target for retail related roles.

The reasons a Position is considered a dimension are similar to that of the Allowance Type. Specifically, it is because that while storing the Positions is an important part of Employee-related business processes, neither abstract Positions nor individual Employees are processes in themselves. Consider the 'PositionAssignment' fact table which is used to sort Employees into Positions or vice-versa. The dimensions are used to store the base details while the fact tables represent business processes and the more advanced details thereof. In summary, while an Employee and Position are important entities on their own, they do not by themselves represent any processes. It is the Position Assignment fact table which brings them together and represents the process of assigning job roles.

The 'Position' dimension as previously discussed is only associated with the  
'PositionAssignment' fact table

#### Timestamp

The unique thing about this dimension is that it is not based on any existing entity in the operational ER diagram. This dimension was artificially created on the basis that it is often considered common practice to include dates and times in dimensional models. This dimension is used to represent both in the one table. Several different entities already use date and/or time related attributes but in the case of a database, a separate table is not often necessary due to the support of data types. As mentioned, this dimension serves to comply with the common practices used for  
dimensional models.

In the operational database, many tables have timestamp related attributes. However, these are loose attributes governed by data type and not a separate entity. These attributes are still present for the dimensional model but instead, they are split to relevant fact tables which invoke Foreign Keys on the Timestamp dimension. What is interesting is that some of these attributes would only store date while others store both date and time. Without worrying about implementation details, it can still be assumed that formatting and casting can be done. For instance, a timestamp is a timestamp but for date-related attributes, the timestamp can be trimmed to show only the date for output.

A Timestamp is identified with a single ID number. While a composite key across the attributes can still provide unique identification, this would violate the rule that Dimensions cannot have composite keys. Aside for the ID, the Timestamp includes a date, month and year as well as hour, minute, seconds and milliseconds.

The 'Timestamp' dimension is associated with the 'PaymentTime', 'ProductDetails', 'CustomerOrderDetails', 'Shift', 'EmployeeDOB', 'PositionAssignment', 'SupplierOrderDetails', 'QuotationDetails' and 'UserAccount'

#### Quotation

Sometimes before an order is made to a Supplier, the responsible manager or overseeing employee may request an estimation as to how much an order of goods will cost. This is referred to as a Quotation. Between the operational entity and the warehouse dimension, all of the Quotation's attributes are retained. A Quotation is identified using a unique number while also storing an internal description and the number of months this quote is valid for. The warehouse also stores dates for when the quote was made and when it will expire. As these are related to the 'Timestamp' dimension, they are offloaded to the 'QuotationDetails' fact table.

For reasons as to why 'Quotation' is a dimension and not a fact, refer to the 'SupplierOrder' dimension.

The 'Quotation' dimension is associated with the 'SupplierOrderDetails', 'QuotationDetails'  
and 'QuotationItem' fact tables.

### Fact Tables

#### SupplierOrderItem

The 'SupplierOrderItem' fact table is used to represent an individual item within a batch of items ordered from a given Supplier. The depiction of this entity is slightly different between the Database and the Warehouse. In the case of the Database, this dimension is known as 'SupplierOrderDetails' while the 'ProductItem' entity represents the individual item as it is brought from a supplier, stocked in store and later sold to a customer. This fact table serves to merge these two database entities into one single fact table for an item ordered from a supplier.

An entry in the 'SupplierOrderItem' fact table is identified using a composite key consisting of the Supplier Order and the Product in question. In other words, an item can only appear once per order. It is the quantity attribute that indicates how many units are being brought. In addition, the fact table also stores the price per unit this item was brought for.

The reason 'SupplierOrderItem' is depicted as a fact table is because it represents the process of tracking and stocking items which were brought from a supplier. This fact table could assist in queries such as which items were stocked during a given period, which items were ordered at a given time, how many times Office Wizard has worked with

The 'SupplierOrderItem' fact table references the 'SupplierOrder' and 'Product' dimensions

#### SupplierOrderPayment

The 'SupplierOrderPayment' fact table represents a payment made to a Supplier in return for the goods they have supplied. This fact table is used to provide an association between a Supplier Order and the payment made in return for the goods received. As such, it only consists of two attributes. The first attribute refers to the payment while the second refers to the Supplier Order this payment is tied to. The payment reference number is used as the sole Primary Key in order to prevent the same payment from being used twice. This is also to allow the possibility that an order may not be paid off all at once. An order may be paid for in multiple instalments. Hence, an order may be tied to multiple payments

The reason 'SupplierOrderPayment' is depicted as a fact table is in order to isolate the process of paying for a Supplier Order into its own table. While this process may not be used for many queries, it can still be used to verify the fact that an order has been paid for, how many payments were made and the total amount thereof. Keep in mind that the payment amount is associated with the dimension while the timestamp is located in a separate fact table.

The 'SupplierOrderPayment' fact table references the 'SupplierOrder' and 'Payment' dimensions

#### Allowance

While the 'AllowanceType' dimension stores the types of allowances that can be granted, the 'Allowance' fact table is used to represent the process of granting an allowance itself. In addition to the normal monthly pay, an Employee can be granted allowances such as for good performance, yearly bonus or sick leave. A given allowance grant is identified with a single ID number. Foreign keys are invoked for the employee receiving the allowance, the type of allowance being granted, the reference number of the allowance payment and the tax bracket being applied to this grant. In addition, the fact table stores attributes for the base allowance amount, the amount subtracted due to tax and the net amount that the employee receives. Also consider that the payment reference number is not stored in a separate fact table in this case. This is due to the fact that allowances must be paid all at once in a single payment unlike supplier and/or customer orders which can often involve loans of some form which are paid off overtime and not in a single instance

The reason 'Allowance' is depicted as a fact table is because it simply represents the process of granting an allowance to an employee. This fact table can facilitate queries such as which Employee has been granted allowances, which allowance is the most frequently used, whether an allowance has been granted often enough relative to its frequency and the total amount of money an Employee has received through grants.

The 'Allowance' fact table references the 'AllowanceType', 'Employee', 'Payment'  
and 'TaxBracket' dimensions

#### Payslip

The 'Payslip' entity is used to represent the salary which is given to Employees on a monthly basis as well as the process of paying that salary thereof. This fact table is similar to Allowance in many ways though the main difference is that Payslips occur for a given Employee on a regular basis while Allowances may only happen once or twice a year. The overall structure of the Payslip entity remains largely unchanged from Database to Fact. A Payslip is identified with a single ID number. Foreign keys are invoked for the Employee receiving the payment, the reference number of the payment and the tax bracket which is applied to this payment as the result of income tax. The reason that the payment reference number appears in the main process fact table and not a separate one is the same as the 'Allowance' process. The 'Payslip' fact table also includes the same payment amount attributes as 'Allowance'. These correspond to the gross pay, taxed amount and net pay.

The reason 'Payslip' is depicted as a fact table is because in a similar manner to 'Allowance', it represents the process of a payment being granted to an Employee. This fact table can facilitate queries such as which employees have received their payslips, how much they are paid, the total income tax paid by Office Wizard (Assumes PAYG system) and whether the payslips match the shifts undertaken by the Employee during the given period.

While the database ER diagram tries to associate Payslip and Shift, the two processes are not directly associated with one another. Indeed, the Shifts undertaken will affect the results of a Payslip but a Shift does not directly result in a Payslip. While a Shift represents the work undertaken for the day, the Payslip represents the total amount paid for the given month. In the database, this would be calculated by a lookup querying the Shift table rather than using a direct Foreign Key.

The 'Payslip' fact table references the 'Employee', 'Payment' and 'TaxBracket' dimensions

#### CustomerOrderPayment

This fact table is similar to 'SupplierOrderPayment'. The main difference is that this fact table is associated with the order made by a Customer rather than an order made to a Supplier. As such, the structure and attributes used remain unchanged. The only real difference is what fact tables and processes the Customer Order Payment is associated with.

For more information regarding the attributes, references and possible queries,  
refer to 'SupplierOrderPayment'.

The 'CustomerOrderPayment' fact table references the 'CustomerOrder' and 'Payment' dimensions. This is exactly similar to the Supplier Order Payment except that the overall process refers to a customer buying goods and not Office Wizard ordering stock.

#### PaymentTime

This fact table is used to associate a given payment with the timestamp in which it occurred. While the 'Timestamp' dimension was created artificially to store time data, this fact table is used to associate a payment with the time it occurred. Therefore, this is not a fact table in the sense that it represents a process as a whole. It is used to comply with both the common practice of depicting time as a dimension while maintaining a functional star schema.

The 'PaymentTime' fact table has only two attributes. The first one for the payment reference number and the second one is for the timestamp number. The payment reference number is used as the sole Primary Key because this fact table is an extension of the base dimension. A payment only occurs once at a given time. While the original database version of the Payment entity stored the equivalent attribute as a date, it makes just as much sense to store a payment as a timestamp. As mentioned in the 'Timestamp' dimension, a full timestamp can be formatted as a mere date if need be. Unlike most fact tables, 'PaymentTime' does not have many possibilities in assisting queries aside for checking when a payment took place. However, it can indirectly aid other payment related queries such as checking the last time a Customer has paid an installment on an order loan.

The 'PaymentTime' fact table references the 'Payment' and 'Timestamp' dimensions.

#### CustomerOrderItem

While the 'SupplierOrderItem' fact table contains information about items ordered in supplier orders, the 'CustomerOrderItem' fact table is used to store information about individual items brought by customers in a given transaction. This fact table is similar to 'SupplierOrderItem' except that it is associated with the process of customers buying goods and not Office Wizard ordering items from suppliers. The two fact tables have the same basic sets of attributes, classification reasons and possible queries. For more information about each, refer to 'SupplierOrderItem'

The 'CustomerOrderItem' fact table references the 'CustomerOrder' and 'Product' dimensions.

#### ProductDetails

Despite its name, the 'ProductDetails' fact table is quite different in purpose from tables such as 'SupplierOrderDetails'. While the latter is used to describe dimensional information about supplier orders, this fact table is related to the Product and is used to sort and categorize the Product by category and/or manufacturer. As mentioned in the 'Category' and 'Manufacturer' dimensions, their depiction as separate entities has not been embraced in the operational database. However, they have been depicted as dimensions in this model so that a Product can be categorized by these dimensions. This sorting can yield a benefit to queries related to products such as which manufacturing organization produces which products and how different products are sorted into categories. This composes the main basic reason as to why 'ProductDetails' is depicted as a fact table. Consider how the 'Product', 'Category' and 'Manufacturer' dimensions fall together into this fact table. When they all come together into one structure, a product has been sorted accordingly.

An entry in the 'ProductDetails' fact table is solely identified with a Timestamp. Hence, the fact table references the 'Timestamp' dimension. As mentioned, the fact table also references the  
'Product', 'Manufacturer' and 'Category' dimensions

#### Shift

The 'Shift' fact table is used to represent the shifts that employees perform their duties while working at Office Wizard. This fact table stems directly from the database ER without the need for additional dimensions supporting different parts of the Shift process. As such, the 'Shift' fact table represents the process of an Employee working and later contributes to the process of calculating an Employee's payslip. Similar to the original entity, a Shift is identified with a single unique number. The fact table includes reference to the Employee working the shift. The 'Timestamp' dimension is also referenced twice as both the shift's starting and ending timestamps. The 'Shift' fact table stores an additional attribute related to the number of hours worked based on the timestamps. In the original ER, the 'Shift' entity included a subclass for shifts worked by sales employees. This subclass has been merged into the superclass for the fact table which stores information on the total amount made in sales as well as how much was discounted from those sales.

The reason that the 'Shift' is depicted as a fact table is because it explicitly represents the act of an Employee performing their duties. This fact table can facilitate queries such as when a given Employee was working, how often they have worked and whether they have missed a shift. Furthermore, the 'Shift' entity as a whole contributes to the calculation of Payslips. Therefore, the fact table can assist the production of payslips and related queries.

The 'Shift' fact table references the 'Employee' and 'Timestamp' dimensions

#### EmployeeDOB

The 'EmployeeDOB' fact table is quite similar to the 'PaymentTime' fact table. They both serve to associate a timestamp with a given dimension. This is due to the fact that since Timestamps are considered a dimension in this model, a separate fact table must be made to associate the timestamp with the dimension without breaking the star schema. The 'EmployeeDOB' fact table only contains two attributes. One of them represents the employee while the other stores that Employee's Date of Birth through the reference of a Timestamp. The Employee ID number is used as the sole Primary Key for this table as this fact table is an extension of the Employee dimension and an Employee can only have one Date of Birth. For more information related to why 'EmployeeDOB' is a fact table, refer to the 'PaymentTime' fact table.

Another similarity between the tables is that neither of them can be used for many queries. The only query that this fact table can really support is finding an Employee's age. It could perhaps be used to find the average age of past Employees or which age demographic has the highest employment rate. Aside for those queries, this fact table cannot contribute much to important processes.

The 'EmployeeDOB' fact table references the 'Employee' and 'Timestamp' dimensions.

#### PositionAssignment

This fact table is used to represent the process of assigning a position or job role to a given Employee. Where the 'Employee' dimension represents an individual Employee and the 'Position' dimension represents a job role, this is the fact table where those two dimensions come together. Between the database design and dimensional model, the attributes stored in a position assignment are mostly identical. The only difference here is how an assignment is identified in this model. In the database, the assignment was identified by a sequential number per Employee. Here, this composite key has been merged into a single number. In addition, attributes are also stored for the position assigned, the starting date and ending date.

The reason why this is depicted as a fact table is because it solely represents the process of assigning Employees to Positions or vice-versa. This could assist in queries such as an Employee's current position as well as what positions they have held in the past or during a given period.

The 'PositionAssignment' fact table references the 'Employee' and 'Position' dimensions.  
The 'Timestamp' dimension is also used twice as the starting and ending dates.

#### SupplierOrderDetails

The 'SupplierOrderDetails' fact table is used to represent the associative dimensions which contribute to the process of ordering goods from a Supplier. While the 'SupplierOrder' dimension indicates the basic existence, this fact table elaborates on that by specifying which Supplier is being given the order, which Employee is responsible for making the order and what Quotation this particular Supplier Order evolved from. In addition, time related attributes are also stored corresponding to the dates that the order was made and received.

An entry in the 'SupplierOrderDetails' fact table is solely identified by the Supplier Order number which references the 'SupplierOrder' dimension. The reason a non-composite Primary Key is used here is because this fact table serves as an extension to the 'SupplierOrder' dimension. In other words, a Supplier Order may only have one corresponding details instance.

The reason why tables of this nature are classified as facts rather than dimensions is to ensure maximum compliance with the star schema. If the 'SupplierOrder' dimension stores the basic existence, this fact table seeks to elaborate on that. The queries that this fact table could support include which suppliers have fulfilled which orders, if a given quotation has directly evolved into an order, how long it takes for orders to be fulfilled and which Employees have communicated with a given Supplier.

The 'SupplierOrderDetails' fact table references the 'Supplier', 'SupplierOrder', 'Quotation', 'Employee' and 'Timestamp' dimensions. For more information on how fact tables of this nature have been modelled, refer to the 'CustomerOrderDetails' fact table.

#### QuotationDetails

In a similar manner to how the 'SupplierOrderDetails' fact table above associates dimensional information with a Supplier Order, the 'QuotationDetails' fact table associates this relevant information with a Quote on Supplier goods. Hence, this fact table contributes to the process of requesting a cost estimate of Supplier goods before ordering them outright. The 'QuotationDetails' fact table stores information regarding which Quotation it is associated with, which Supplier is being asked to make the quotation and which Employee has requested this quotation. Time related attributes are also stored which depict the date of the quotation request and when it is set to expire. Its expiry date is based on the 'validPeriod' attribute within the 'Quotation' dimension.

For information regarding the 'QuotationDetails' Primary Key and why it is a fact table,  
refer to 'SupplierOrderDetails'

The 'QuotationDetails' fact table is associated with the 'Quotation', 'Supplier', 'Employee'  
and 'Timestamp' dimensions.

#### QuotationItem

Where the 'QuotationDetails' fact table above describes the dimensional details of a quotation, the 'QuotationItem' fact table stores the individual products that are requested for quotation from the Supplier. As the name may suggest, this fact table is similar in purpose to the 'CustomerOrderItem' and 'SupplierOrderItem' fact tables. The main difference is that these processes relate to the direct buying and selling of goods. A quotation is only the process of a hypothetical supplier order in order to gauge the approximate price for a set of goods. Therefore, the 'QuotationItem' fact table stores an individual item being estimated for a quotation.

Despite these differences between the processes, the set of attributes is functionally identical between all three item related fact tables. However, the 'price' attribute in the 'QuotationItem' fact table has been renamed to 'priceQuoted', meaning that the price is only the estimate in the quotation and not the amount passed over a transaction.

For more information regarding why 'QuotationItem' is a fact table and how it is identified,  
refer to the 'SupplierOrderItem' fact table. The possible queries are also quite similar but keep in mind that a quotation is not a transactional event. It is only a simulation of how a similar transaction may proceed. Nonetheless, this fact table can facilitate queries related to comparing the prices stated in quotations to the actual prices the goods are brought for when ordering.

The 'QuotationItem' fact table references the 'Quotation' and 'Product' dimensions.

#### UserAccount

As part of Office Wizard's computer system, it is a requirement that all Employees have a corresponding user account in order to access this system. The dimensional model also complies with this rule as the 'UserAccount' fact table is used to store the user account information of all Employees in the business. As the User Account was a simple table in the Database, its fact table equivalent depicts all of the attributes unchanged. However, the one major difference is that the table also includes a reference to the Timestamp dimension to serve as part of the composite key. This is used to enable the storage of both past and present Employee credentials. The composite key consists of the Employee ID and the timestamp indicating the credentals instance.

For the basic reason why 'UserAccount' refers to fact table, refer to 'EmployeeDOB'. Both fact tables serve as an extension of the Employee but this time, 'UserAccount' serves much more of an overall purpose in the system. In effect, it could serve as an index to facilitate queries related to user accounts. For example, assume that Employees can change both their username and password. The fact table would facilitate a query to check which Employees have held a given username during a given time. It would also assist in retrieving past passwords used for security related processes. Therefore, this fact table serves as more of an index than as part of any real process

The 'UserAccount' fact table references the 'Employee' and 'Timestamp' dimensions.

## Dimensional Matrix

Now that all of the different tables have been explained and reasoned with, it is time to outline the dimensional matrix. When creating an ER diagram for an operational database, one may choose to include a relationship table alongside. Since this model is for a data warehouse and not an operational database, providing a dimensional matrix grid instead would be the more suitable option. This grid is also commonly referred to as a “Data Warehouse Bus Matrix”.

According to the Katie and Emil academy, the dimensional or bus matrix is used to present the relations between business processes and dimensions. By creating this relationship grid, it is much easier to ensure a common understanding of the warehouse's objective between both technical and business people.

In this case, it is not important to know the theory of making a dimensional matrix. All that matters is that on the grid, the dimension tables are listed on one axis and the fact tables are listed on the other. If there is a mark in the cell where a dimension and fact intersect, that means there is a relationship between them.

The Dimensional Matrix for this particular model was created using Google Sheets. The benefit of using a spreadsheet instead of embedding a table directly into the document is that the spreadsheet is orientation-independent. As a word processing document is often portrait by default, it becomes tricky to embed tables that carry more width and length while maintaining readability. Instead, the matrix was created in the spreadsheet, exported as a PDF, converted to an image and then embedded into the document. If for some reason an image needs to be displayed in landscape instead, it only needs to be rotated and resized which is a lot easier.

The exported Dimensional Matrix has been embedded as Appendix 3.

## EER Diagram

While a dimensional matrix was used in place of the relationship table in this case, the familiar EER diagram has remained the same. For this design, an EER diagram was developed in order to illustrate the facts, dimensions and how they all connect together. It also outlines the attributes stored within each table and which of these attributes function as keys.

The EER diagram was modelled entirely using Draw.IO. This is different in comparison to the operational database EER which was drafted using Draw.IO and finalized using Microsoft Visio. Aside for the tools used, the diagram was exported the same way as the Dimensional Matrix grid.

The EER diagram has been embedded as Appendix 4

### Notation

Although the systems are different, the EER diagram for the dimensional model uses the same notation as operational database EER diagrams. Therefore, if you understand a database EER, you should have no problem understanding the warehouse EER which is the same for the most part. However, the diagram uses some non-standard notation that should be explained in this section.

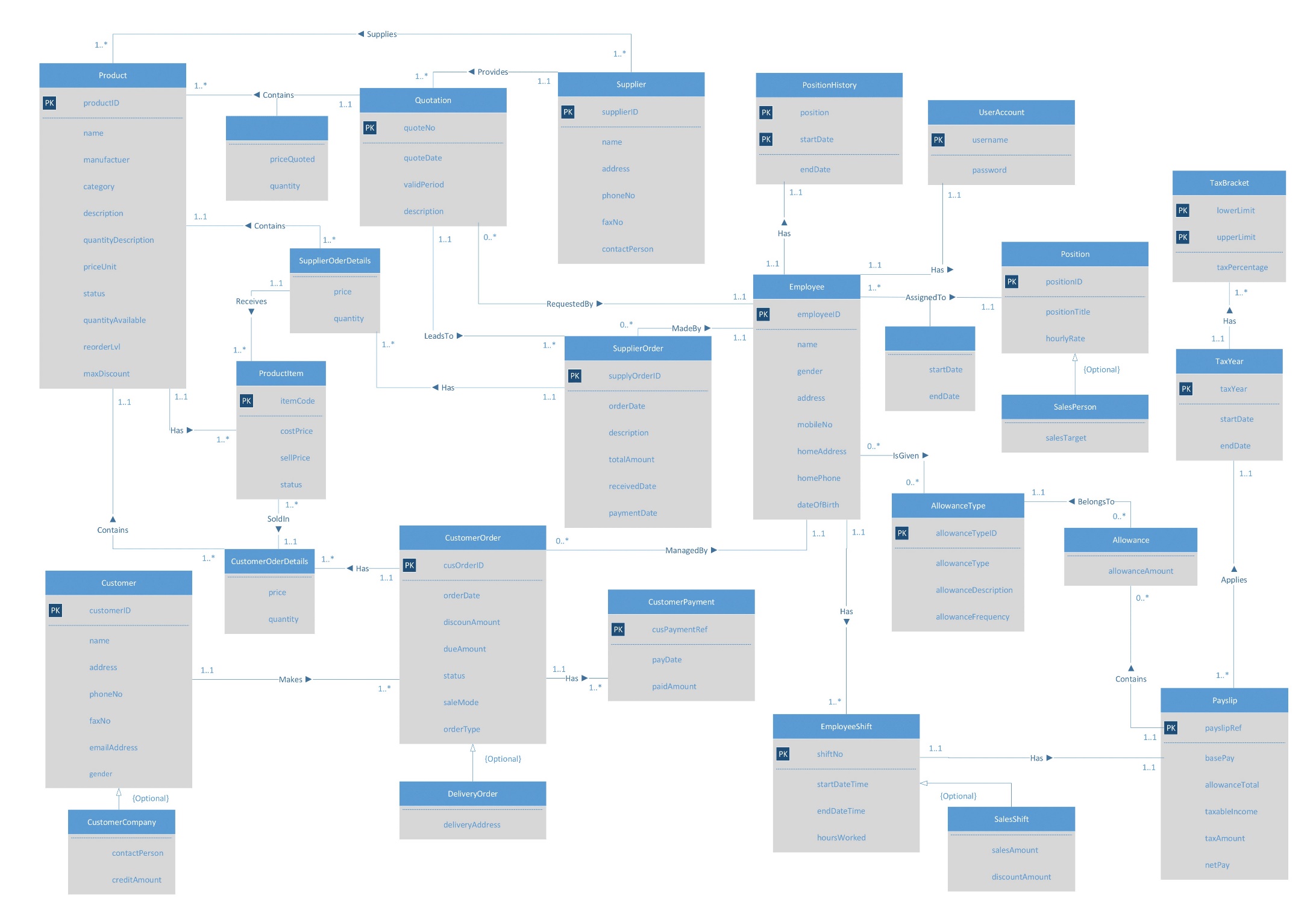
First of all, to clarify which tables are fact or dimension, the table objects in the diagram have been coloured accordingly. Dimension tables are coloured as Blue while Fact tables are coloured as Green. It is also important to note that in foreign key instances, the parent attributes also appear in the child table. Some people prefer not to include them due to the fact their presence should be implicit and do not need to be reinforced. Regardless, it still adds clarity to how the database might look when it is implemented.

The second non-standard part of the diagram’s notation is that you will quickly notice coloured squares sitting between the different tables with lines coming in and out. These coloured squares are used to direct connections to destination table without cluttering the diagram with individual lines to represent individual relationships. For example, if two dimensions associated with the same facts are sitting adjacent, they might individually connect to a Blue square. From there, the connection is merged where it will connect to a Green square and onto the target fact table. If you have trouble understanding this, pick a given relationship and trace the lines from  
dimension to fact(s)

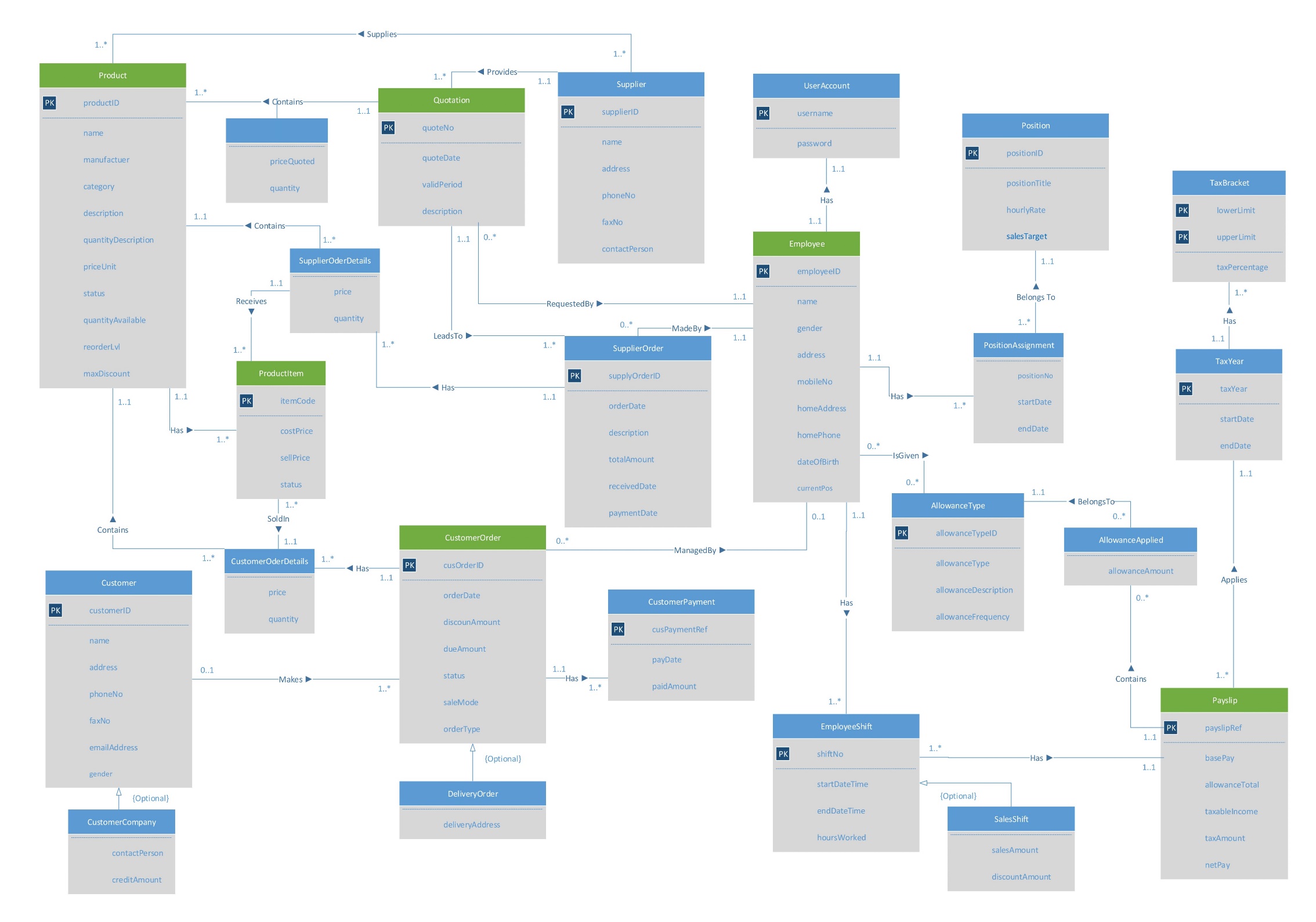
The squares are coloured with the same scheme as the tables themselves. If the square is Blue, that means it is associated with connected dimension tables and only routes connections for these tables. If the square is Green, that means it relates to a fact table. The Green squares are used to merge the incoming dimensions into their target fact table. To reiterate, if you have trouble understanding, follow the lines.

## Appendices

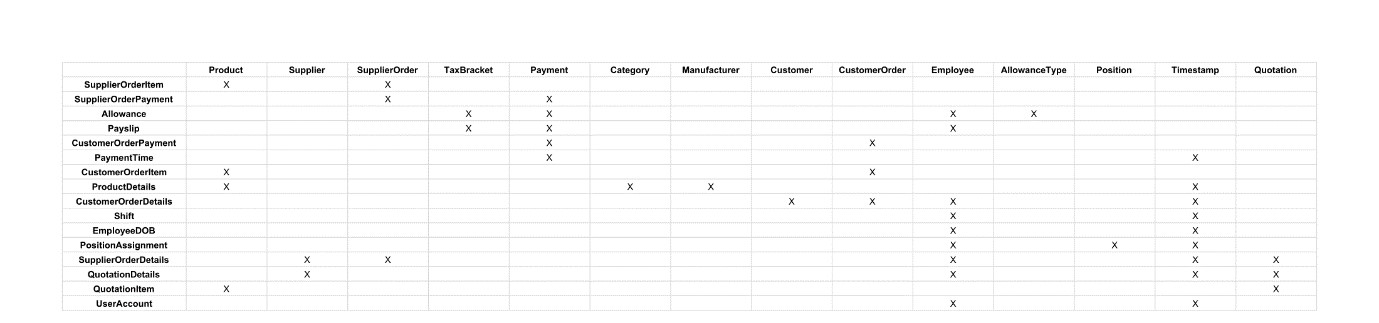
### Appendix 1 – Operational EER Diagram



### Appendix 2 – Revised Schema EER Diagram



### Appendix 3 – Warehouse Bus Dimensional Matrix



### Appendix 4 – Dimensional Model EER Diagram

## References

* 'Database Systems: A Practical Approach to Design, Implementation and Management'.  
  6th Edition (Global).  
  Written by Thomas M. Connolly and Carolyn E. Begg.  
  Published by Pearson Education Limited.  
  Pages 1260-63.
* Katie and Emil academy Business Intelligence educational blog.  
  'Kimball Bus Matrix'.  
  Written by Emil Glownia.  
  Published on 20th November 2014.  
  (<https://katieandemil.academy/kimball-bus-matrix-in-data-warehouse>)
* PDF 2 PNG.  
  Conversion website used to convert .pdf files into .png images  
  Used to convert diagram files into better quality images for easier embedding  
  (<http://pdf2png.com/>)
* Draw.IO  
  Browser-based diagramming application  
  Used to create Dimensional Model EER diagram  
  (<https://www.draw.io/>)
* Google Sheets  
  Browser-based spreadsheet application provided by Google  
  Part of overall ‘Google Docs’ service  
  Used to create Dimensional Matrix grid  
  (<https://www.google.com.au/sheets/about/>)