My Final Project Of Relational Database-management System (MS-DS Program On Coursera)

Back-ground Story

My client Quantum-Apex
is one of the fastest and rapidly growing , renowned e-commerce platforms and also one of the largest Shopping Mall Of hardware products that offers a wide range of products and services to the customers around the globe with a strong reputation in the industry offering exceptional and unique global digital customer service and delivering top-quality hardware products, which has rewarded them as a Sustained and committed customer base . Recently in their 12'th anniversary , Their aim is to create a contemporary and Flexible Data infrastructure model capable of managing the expanding data volume and accommodating the development of their business.Because in a survey , many employees claim that they are having great difficulty in handling the growing amount of data, which is causing their data queries and reports to take longer to process and On occasion, the system may generate inaccurate outcomes.
. An employee named Roumi recently connected with me on the ZOOM platform and explained to me the whole issue. As a database management expert, I have been asked to build an RDBMS system for building a modern digital platform and making a scalable database management system so that they can support their company's growing business and reputation . My objective was to ensure that their system can accommodate their company's expanding business needs, while also providing accurate, reliable, and easily accessible data in their digital online platform.

Making an Entity Relationship Model:

Entities

After discussion for an hour , We together agree these entities

Purchaser: DeliveryID, FirstName, LastName, Address, Email, Phone, Negotiation, Payment-type,Satisfaction:. **Here DeliveryID is the identifier**.

Products : DeliveryID , Product-name#, Brand,Category,Description,Price:,Availability,Features,Reviews.**Here DeliveryID and Product-name# together as the identifier.**

Client meeting: MeetingtID, Date, Time,Platform-Name ,Registration cost, Rate given by customer . **Here MeetingID is the identifier**

Staff: ServiceID, FirstName, LastName, Biometric Data, Address, Email, Phone,Company Join Date, administratorID.Here **ServiceID is the identifier.**

Relations

THEIR RELATIONSHIPS ARE:

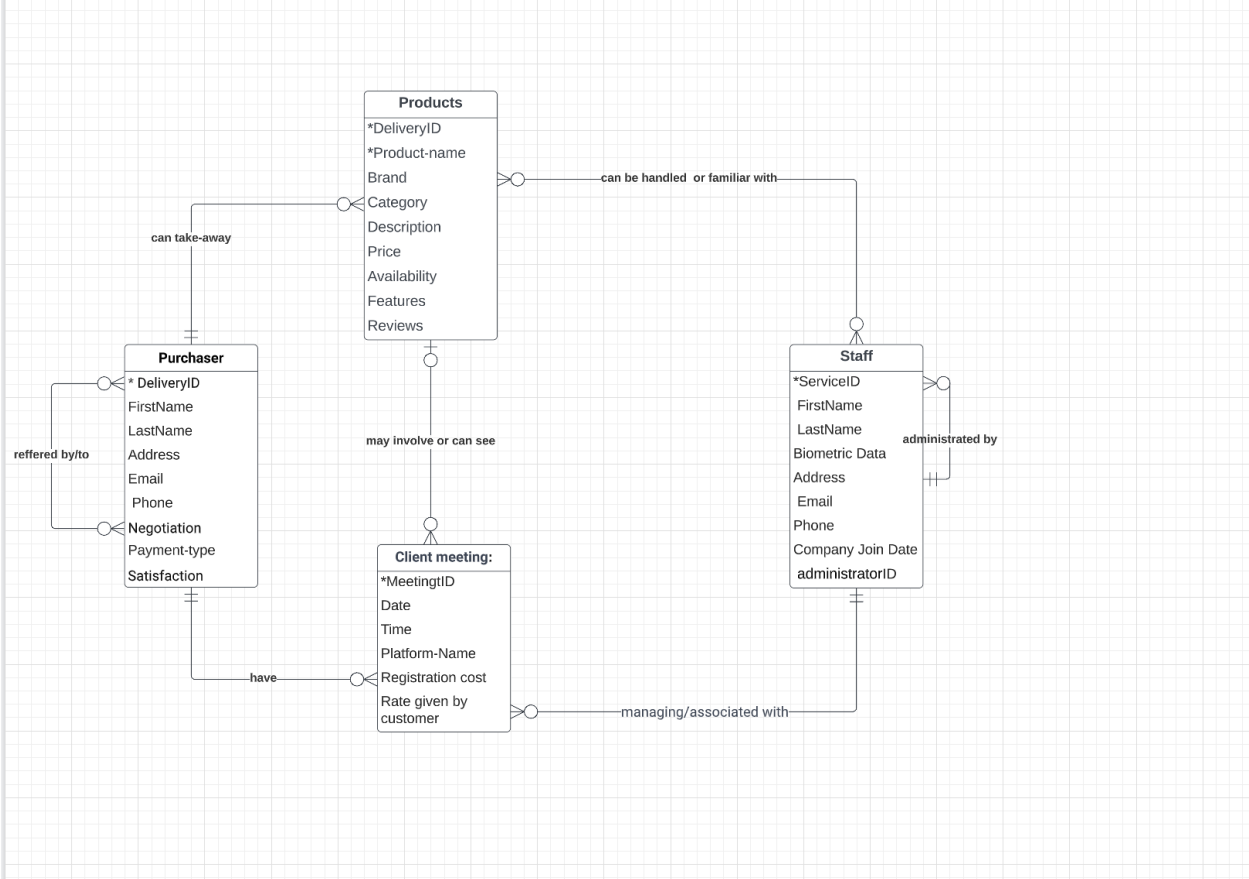
- A purchaser may buy one or more (zero or more) products**(an example of a one-to-many (1:N) relationship).**; A product must belong to one and only one (exactly one) purchaser(an example of N:1 relation)...
- Each staff member can handle zero or more(1:N) products, while each product can be handled by one or more staff members.**(an example of a many-to-many (N:M) relationship)**
- There is a one-to-many (1:N) relationship between purchaser and visits, where each purchaser can have zero or more visits. Additionally, there is a many-to-one (N:1) relationship between visits and purchasers, where each visit is associated with one and only one purchaser.
- Products may have one or more (zero or more)purchasers- visits; and each purchaser visit may involve or can see one (zero or one) product .
- There is a one-to-many (1:N) relationship between staff and Client , where each staff member can be associated with one or more client meetings . Additionally, there is a many-to-one (N:1) relationship between client meeting and staff, where each client meeting must be proceeded by one and only one staff member.
- A purchaser may be referred by one or more (zero or more) other purchasers; and a purchaser may be referred to one or more (zero or more) purchasers. It is like an Unary:Many to Many Relationships!

- A staff member must have one and only one (exactly one) administrator; A staff may be administered by one or more (zero or more) staff.It is also Unary : 1 to many relationship

:

Creating the Entity Relationship Diagram(ERD)

Please turn over



Converting the ERD to the Relational Model:

Purchaser: DeliveryID, FirstName, LastName, Address, Email, Phone, Negotiation, Payment-type,Satisfaction,ReferredByDeliveryID(fk))

Products : DeliveryID(fk) , Product-name#, Brand,Category,Description,Price:,Availability,Features,Reviews.

Client meeting: MeetingtID, Date, Time,Platform-Name(fk) ,Registration cost(fk), Rate given by customer(fk) . **Here MeetingID is the identifier**

Staff: ServiceID, FirstName, LastName, Biometric Data, Address, Email, Phone,Company Join Date, administratorID(fk)**.**

Products_Staff(DeliveryID(fk), Product-name#(fk),ServiceID(fk))

Normalizing the Relational Model to 3NF

QUANTUM APEX assists me by telling functional dependencies of the relational Model to convert it in 3NF

Purchaser(DeliveryID, FirstName, LastName, Address, Email, Phone, Negotiation, Payment-type,Satisfaction,ReferredByDeliveryID(fk)).

- **FD1**:DeliveryID → FirstName, LastName, Address, Email, Phone, Negotiation, Payment-type,Satisfaction,ReferredByDeliveryID(fk)

Products (DeliveryID(fk) , Product-name#, Brand,Category,Description,Price:,Availability,Features,Reviews)

FD1: DeliveryID ,Product-name → Brand,Category,Brand-Description,Price:,Availability,Features,Reviews

FD2: DeliveryID → Category,Features,,Availability

FD3: Brand → Review, Brand Description

Staff(ServiceID, FirstName, LastName, Biometric Data, Address, Email, Phone,Company Join Date, administratorID(fk)**.**

- **FD1:** ServiceID → FirstName, LastName, Biometric Data, Address, Email, Phone,Company Join Date, administratorID(fk)

Client meeting: MeetingtID, Date, Time,Platform-Name(fk) ,Registration cost(fk), Rate given by customer(fk)

- **FD1:** MeetingtID → Date, Time,Platform-Name,Registration cost, Rate given by customer
- **FD2**: Rate Given By Customer →Platform-Name(fk), Registration cost(fk),Time

- **Product-name_Staff**(DeliveryID(fk), Product-name#(fk), MeetingtID(fk))
  - The attribute that serves as the primary key is not present in this context!

Here is the normalization process:

- ☐ **Purchaser, Staff, and Product-name_Staff relations are in 3NF, because they are in 1NF and they have no :-**
- ☑ ~~**partial functional dependencies**~~

☑ ~~no transitive functional dependencies~~

So, they are in 2NF and as well as 3 NF

☐ **Products relation is in 1NF**
☐ **They are not in 2 NF because**
☐ They have partial functional dependencies like DeliveryID → Category,Features,,Availability

-> Create a new relation to put DeliveryID,Category,Features,,Availability. Since deliver id relation has these attributes, we can simply remove them from product, and keep DeliverID as a foreign key.

->**Products** (DeliveryID(fk) , Product-name#, Brand,Category,Description,Price:,Availability,Features,Reviews)

 # FD1: DeliveryID ,Product-name → Brand,Category,Brand-Description,Price:,Availability,Features,Reviews

FD2: Brand → Review, Brand Description

Products relation now is in 2NF. However, it is not in 3NF because of FD2: Brand → Review, Brand Description (DeliveryID, Product-name#) →Brand, and Brand->is a transitive functional dependency. We need to normalize Products to 3NF:

- Make a new relation to put Review, Brand Description and modify Product:
- Brand (Review,Brand-Description)
  - FD1: Brand → Review, Brand Description

Products (DeliveryID(fk) , Product-name#, Brand(fk),Category,Description,Price:,Availability,Features,Reviews).

- FD1: DeliveryID(fk) , Product-name# -> Brand,Category,Description,Price:,Availability,Features,Reviews
- Now Brand relation is in 3NF

- ☐ Customer Visit relation is in 1NF, and 2NF. But it is not in 3NF because of functional dependencies and transitive dependencies : N.B: ServiceDescription is a transitive functional dependency.

  By following same procedure we get

Client meeting:

MeetingtID -> Date, Time,Platform-Name(fk) ,Registration cost(fk), Rate given by customer(fk)

FINAL OUTCOME:

Purchaser(DeliveryID, FirstName, LastName, Address, Email, Phone, Negotiation, Payment-type,Satisfaction,ReferredByDeliveryID(fk)).

- **FD1**::DeliveryID → FirstName, LastName, Address, Email, Phone, Negotiation, Payment-type,Satisfaction,ReferredByDeliveryID(fk)

Products (DeliveryID(fk) , Product-name#, Brand,Category,Description,Price:,Availability,Features,Reviews)

- **FD1**::DeliveryID(fk) , Product-name# -> Brand,Category,Description,Price:,Availability,Features,Reviews

Brand (Brand, Brand Description)

- **FD1**: Brand→ Brand Description

Staff(ServiceID, FirstName, LastName, Biometric Data, Address, Email, Phone,Company Join Date, administratorID(fk)**.**

- **FD1:** ServiceID → FirstName, LastName, Biometric Data, Address, Email, Phone,Company Join Date, administratorID(fk)

Client meeting(MeetingtID, Date, Time,Platform-Name(fk) ,Registration cost(fk), Rate given by customer(fk))

- **FD1:** MeetingtID→ Date, Time,Platform-Name ,Registration cost, Rate given by customer

- **Service-Rate** (Rate Given By Customer,Platform-Name(fk), Registration cost(fk),Time
-
  - **FD1**: Rate Given By Customer →Platform-Name, Registration cost,Time

- **Product-name_Staff**(DeliveryID(fk), Product-name#(fk), MeetingtID(fk))
    - The attribute that serves as the primary key is not present in this context!

Now I finished my Relational Data-base Management Final Project and Now I need to make a video presentation!!

Citation/Reference::

1)University Of COlorado,Boulder (n.d.). *Coursera MS-DS Program Relational Database Management System Sample Project , week4*. Coursera. https://www.coursera.org/learn/relational-database-design/home/week/4

2)N. Z. (n.d.). *Data Warehousing for Business Intelligence*. Github. https://github.com/nickzarate/database-management-essentials