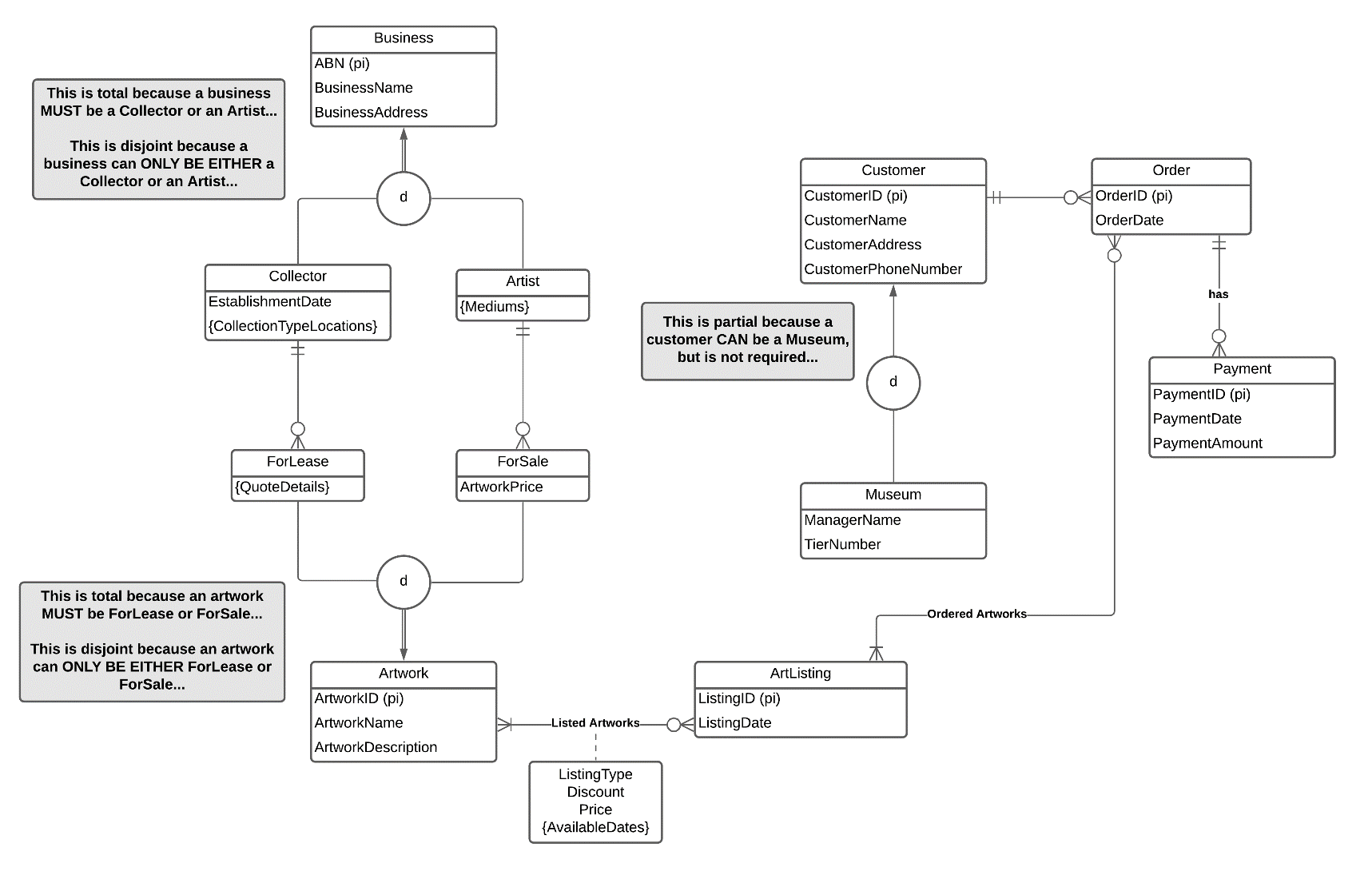
## **Final Exam- Comp1350, S2 2021**

|  |  |
| --- | --- |
| Student ID | 45188416 |
| Student Name | Jai Carey |
| Subject Code | COMP1350 |

Section-A

1. Question1
   1. No central repository for data which can cause challenges with collaboration and limits data sharing.
   2. Each department only has access to their own data so can only make decisions with what they see and will miss opportunities and trends.
   3. This scenario can cause data duplication which increases costs and reduces system efficiency.
   4. If the business were to add a new department, an entire new file system and application would need to be installed as existing infrastructure is never reused.
   5. Employees moving between departments or trying to use another departments system would require additional training which increases costs and reduces collaboration.
2. EER Diagram:

Assumptions:

* “Type of collections locations” means that Collectors can have many different collections of art and each collection can be at their own location
* Only 1 artist can sell an artwork (multiple artists do not work together and sell 1 painting they all worked on – they work alone)
* Collectors ONLY lease artworks, they do not also try to sell and lease artworks
* Artwork description is not needed in the attribute linked to Listed Artworks as this is already stored against the Artwork
* Artworks are ordered against listed artworks, the ArtwordID is passed through the ArtListing to ensure this
* Regardless of future plans of selling different types of artwork, we only designed this EER diagram based on current business / sales model
* ListingType apart of the attribute linked to the Listed Artworks refers to if it is for sale or lease
* An ArtListing must have ATLEAST 1 Artwork, otherwise there is no point to the ArtListing as there is nothing to display…

Explanation of concepts:

* 1. 1. **Strong Entity** – Business (ABN (PK), BusinessName, BusinessAddress)
     2. **Weak Entity** – Payment (OrderID (PK, FK), PaymentID (PK), PaymentDate, PaymentAmount)
  2. **Associative Entity** - Attribute on Many to Many relationship ListedArtworks (ListingType, Discount, Price, {AvailableDates})
  3. **Composite Attribute** - Customer Name which would include the First and Last name and could be broken down further
  4. **Multi-Valued Attribute** – The attribute in Artist, Mediums
  5. 1. **Total Constraint** – Between the disjoint and Business
     2. **Partial Constraint** – Between the disjoint and Customer

1. Logical Transformation:

**Step 1 (Strong Entities):**

* BUSINESS (**ABN (PK)**, BusinessName, BusinessAddress)
* ARTWORK (**ArtworkID (PK)**, ArtworkName, ArtworkDescription)
* ARTLISTING (**ListingID (PK)**, ListingDate)
* CUSTOMER (**CustomerID (PK)**, CustomerName, CustomerAddress, CustomerPhoneNumber)
* ORDER (**OrderID (PK)**, OrderDate)

**Step 2 (Weak Entities):**

* PAYMENT (**OrderID (PK, FK)**, **PaymentID (PK)**, PaymentDate, PaymentAmount)

**Step 3 (1 to 1 Relationships):**

* There are no 1 to 1 relationships

**Step 4 (1 to Many Relationships):**

* ORDER (**OrderID (PK)**, OrderDate, **CustomerID (FK)**)
* PAYMENT (**OrderID (PK, FK)**, **PaymentID (PK)**, PaymentDate, PaymentAmount)

**Step 5 (Many to Many Relationships):**

* LISTED ARTWORKS (**ArtworkID, ListingID,** ListingType, Discount, Price)
* ORDERED ARTWORKS (**ArtworkID, OrderID**)

**Step 6 (Multi-Valued Attributes):**

* AVAILABLE DATES (**ArtworkID, ListingID**, AvailableDate)

**Step 7 (Associative Entities):**

* There are no associative entities

**Step 8A (Superclass and Subclasses):**

* COLLECTOR (**ABN (PK)**, EstablishmentDate)
* ARTIST (**ABN (PK)**)
* FORLEASE (**ArtworkID (PK)**)
* FORSALE (**ArtworkID (PK)**, ArtworkPrice)
* MUSEUM (**CustomerID (PK)**, ManagerName, TierNumber)

**Repeat Steps 2-7:**

**Step 2 (Weak Entities):**

* No Weak Entities

**Step 3 (1 to 1 Relationships):**

* No 1 to 1 Relationships

**Step 4 (1 to Many Relationships):**

* FORLEASE (**ArtworkID (PK), CollectorABN (FK)**)
* FORSALE (**ArtworkID (PK)** , ArtworkPrice, **ArtistABN (FK)**)

**Step 5 (Many to Many Relationships):**

* No Many to Many Relationships

**Step 6 (Multi-Valued Attributes):**

* COLLECTION TYPE LOCATIONS (**ABN (PK)**, CollectionType, CollectionLocation)
* MEDIUMS (**ABN (PK)**, ArtMedium)
* QUOTE DETAILS (**ArtworkID (PK)**, TotalQuotePrice, QuoteDate, OptionalInsurancePrice)

**Step 7 (Associative Entities):**

* No Associative Entities

**Final List**:

* BUSINESS (**ABN (PK)**, BusinessName, BusinessAddress)
* ARTWORK (**ArtworkID (PK)**, ArtworkName, ArtworkDescription)
* ARTLISTING (**ListingID (PK)**, ListingDate)
* CUSTOMER (**CustomerID (PK)**, CustomerName, CustomerAddress, CustomerPhoneNumber)
* ORDER (**OrderID (PK)**, OrderDate, **CustomerID (FK)**)
* PAYMENT (**OrderID (PK, FK)**, **PaymentID (PK)**, PaymentDate, PaymentAmount)
* LISTED ARTWORKS (**ArtworkID, ListingID,** ListingType, Discount, Price)
* ORDERED ARTWORKS (**ArtworkID, OrderID**)
* AVAILABLE DATES (**ArtworkID, ListingID**, AvailableDate)
* COLLECTOR (**ABN (PK)**, EstablishmentDate)
* ARTIST (**ABN (PK)**)
* FORLEASE (**ArtworkID (PK), CollectorABN (FK)**)
* FORSALE (**ArtworkID (PK)** , ArtworkPrice, **ArtistABN (FK)**)
* MUSEUM (**CustomerID (PK)**, ManagerName, TierNumber)
* COLLECTION TYPE LOCATIONS (**ABN (PK)**, CollectionType, CollectionLocation)
* MEDIUMS (**ABN (PK)**, ArtMedium)
* QUOTE DETAILS (**ArtworkID (PK)**, TotalQuotePrice, QuoteDate, OptionalInsurancePrice)

**Step 8B:**

* COLLECTOR (**ABN (PK)**, BusinessName, BusinessAddress, EstablishmentDate)
* ARTIST (**ABN (PK),** BusinessName, BusinessAddress)
* FORLEASE (**ArtworkID (PK),** ArtworkName, ArtworkDescription)
* FORSALE (**ArtworkID (PK)**, ArtworkName, ArtworkDescription, ArtworkPrice)

**Step 8C:**

* BUSINESS (**ABN (PK)**, BusinessName, BusinessAddress, EstablishmentDate, BusinessType)
* ARTWORK (**ArtworkID (PK)**, ArtworkName, ArtworkDescription, ArtworkPrice, ArtworkType)
* CUSTOMER (**CustomerID (PK)**, CustomerName, CustomerAddress, CustomerPhoneNumber, ManagerName, TierNumber, CustomerType)

**Step 8D:**

* There is no Overlap used

1. 3NF Steps:

**Given the below table is already in 1NF, we first need to change that to 2NF**.

2NF Table Relations

Full Dependency: ArtistID, PaintingID 🡪 ArtistName, ArtistLoc, PaintingName, Price

Partial Dependency 1 (Artists): ArtistID 🡪 ArtistName, ArtistLoc

Partial Dependency 2 (Paintings): PaintingID 🡪 PaintingName, Price

Once we have separated out the Partial Dependencies into their own tables and use them as references, we then look at any Transitive Dependencies and move those into their own tables.

Given our new tables Artists and Paintings, there are no transitive dependencies that we can move into a 3NF form. So therefore, with this case study, 2NF and 3NF are the same outcome.

Section-B (Insert your SQL code under each of the questions)

1. Query 1:

CREATE TABLE Exhibition (

ExhibitionID CHAR(3) NOT NULL,

ExhibitionDate DATE NOT NULL,

ArtistID CHAR(5) NOT NULL,

ExhibitionCost DECIMAL(5, 2),

PRIMARY KEY(ExhibitionID, ExhibitionDate),

FOREIGN KEY(ArtistID) REFERENCES Artist(ArtistID)

);

1. Query 2:

INSERT INTO Artist (ArtistID, ArtistName, ArtistCountry) -- DONE

VALUES ('A1234', 'Jai Carey', 'Australia'); -- DONE

INSERT INTO PaintingType (PaintingTypeID, PaintingTypeName)

VALUES ('PT12', 'Oil Painting');

INSERT INTO Painting (PaintingID, PaintingName, PaintingType, ArtistID)

VALUES ('P12345', 'WOWOWOWOW', 'PT12', 'A1234');

INSERT INTO Exhibition (ExhibitionID, ExhibitionDate, ArtistID, ExhibitionCost)

VALUES ('E23', '2020-05-03', 'A1234', 140.95);

1. Query 3:

SELECT a.ArtistID, a.ArtistName

from Artist a

WHERE a.ArtistCountry LIKE '\_\_g%';

1. Query 4:

SELECT p.PaintingName, pt.PaintingTypeName

FROM Painting p JOIN PaintingType pt

USING (PaintingTypeID);

1. Query 5:

SELECT a.ArtistName, COUNT(p.ArtistID) AS 'Number of Paintings'

FROM Artist a LEFT JOIN Painting p

USING(ArtistID)

GROUP BY a.ArtistName

ORDER BY COUNT(p.ArtistID) DESC;

1. Query 6:

SELECT a.ArtistName, COUNT(p.ArtistID) AS 'Number of Paintings'

FROM Artist a LEFT JOIN Painting p

USING(ArtistID)

GROUP BY a.ArtistName

HAVING COUNT(p.ArtistID) > AVG(COUNT(p.ArtistID))

ORDER BY COUNT(p.ArtistID) DESC;

1. Query 7:

SELECT COUNT(e.ExhibitionID) AS 'Number of Acrylic Exhibitions'

FROM Exhibition e, Painting p, PaintingType, pt

WHERE e.ArtistID = p.ArtistID AND p.PaintingTypeID = pt.PaintingTypeID AND pt.PaintingTypeName LIKE '&acrylic%';

1. Query 8:

SELECT a.ArtistName

FROM Artist a

WHERE ArtistID IN (

SELECT p.ArtistID

FROM Painting p JOIN PaintingType pt

USING(PaintingTypeID)

WHERE pt.PaintingTypeName LIKE '&watercolour&'

);

1. Query 9:

SELECT a.ArtistName

FROM Artist a JOIN Painting p JOIN PaintingType pt

ON a.ArtistID = p.ArtistID AND p.PaintingTypeID = pt.PaintingTypeID AND pt.PaintingTypeName LIKE '&watercolour&';

1. Query 10:

-- ASSUMPTION - 1ST QUARTER REFERS TO THE AUSTRALIAN FINANCIAL CALENDAR (JULY, AUG, SEPT)

SELECT e.\*, a.ArtistName

FROM Exhibition e JOIN Artist a

USING(ArtistID)

WHERE YEAR(e.ExhibitionDate) = 2019

AND MONTH(e.ExhibitionDate) IN (7, 8, 9)

AND e.ExhibitionCost BETWEEN 500 AND 750;

Section-C

1. Question 1:

**Thick-client 1:**

A potential thick-client that could exist in this environment could be a CRM System to manage customer and business relations. A thin-client would be more appropriate for this as the database would need to transfer all data to a client to enable processing on the client, which would be a security and efficiency issue.

**Thick-client 2:**

Another potential thick-client that could exist in this environment could be a COVID sign in system for contact tracing. The storage of the customer and business unique information would occur in the database, but the client would handle all of the identification confirmation against the persons driver’s license or similar photo ID. This would work best as a thick client so that photos of the individual are not sent to and from the server, especially as they are not being stored, and this would cause security issues.

**Thick-client 3:**

The final potential thick-client that could exist in this environment could be Microsoft Outlook and the businesses email system. This would be a thick-client as the database only stored emails, but does not process them or any information. The client would download all of the emails and searches, pagination, filing, viewing or any other activity would be performed by the persons local computer. This is important to allow for formal communication with customer and businesses. This is best as a thick-client.

1. Question 2:

A data-mart is typically a subset of a data warehouse which includes highly organised data for specific functions. Pick-up Arts can use a data-mart to make informed business decisions through the data analytic functionality of their data-marts. They can do this by storing all records such as customers, businesses, artworks, orders, payments and other business processes to use this data and business analysis strategy enabling improved decision making.

**Scenario 1**: Storing order, price and artwork type data can allow Pick-up Arts to perform trend analysis on successful sales during different periods of time and provide their Artists and Collectors insights into how they compare to each other.

**Scenario 2**: Storing the location of leasable artwork and the location of customers can allow Pick-up Arts to find a trend between leasing and distance between the goods and customer upon sale. This can also be used to create new services such as expected travel time to pick-up and even accurately estimate costs for delivery services.

**Scenario 3**: Storing of customer details and customer orders allows for targeted marketing to repeat / loyal customers and the effectiveness of this marketing return on investment can be compared to marketing for cold leads. This allows the business to understand where to spend their marketing budget to maximise their return on investment and more accurately match their budget to the strategic goals.

1. Question 3:

There are 3 key reasons that Instagram might not use a relational database to store all their data. 1 is that relational databases are difficult to use for business and data analytics. 2 is that relational databases are not effective at storing a large portion of Instagram’s data, images, videos and audio. 3 is that as a business begins to scale massively, their data solutions must also scale with them, and relational databases begin to have performance, cost and complexity issues as the database grows.

An alternative solution for the storage of Instagram’s data is to have a multi-storage strategy, which includes file storage for larger files such as videos and a data warehouse for all other data stored.

The data warehouse can be used to store data such as profile information, customer demographics, customer behaviour, advertising trends, content trends and much more. The data warehouse excels as the collection of data from multiple known sources and organising them for efficient data and business analysis allowing for improved and quicker decision making.

With the collected data, and the utility of the data warehouse, Instagram can not only make better decisions for themselves as a business but can use the data warehouse to create new services for their customers, the advertisers. For example, giving them more insight into their expected ROI based on history which has been stored in the data warehouse.