Temasek Polytechnic

School of Informatics & IT

**Database Application Development (CIA1C06)**

**AY 23/24 Oct Semester**

**Project Part 2**

**(Database Design)**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

Table of Contents

**Declaration of Originality** 3

Guidelines for designing ER-Model 4

Part A: The Entity-Relationship Diagram 6

Part B: The Entity-Attributes Information 7

Part C: The Report 8



**SCHOOL OF INFORMATICS & IT**

**[Database Design Document]**

Student Name (Matric Number) : Shubham Kaushik

Tutorial Group : T14

Jenny LING (TP)

Tutor :  **Mr Effendi BABA**

Submission Date : 26/11/23

**Declaration of Originality**

I am the originator of this work and I have appropriately acknowledged all other original sources used as my references for this work.

I understand that Plagiarism is the act of taking and using the whole or any part of another person’s work, including work generated by AI, and presenting it as my own.

I understand that Plagiarism is an academic offence

and if I am found to have committed or abetted the offence of plagiarism in relation to this submitted work, disciplinary action will be enforced.

# Guidelines for designing ER-Model

**Definition**

ER-Model

Entity Relationship model constructs entity sets, relationship sets, and imposes constraints. ER Model is best used for the conceptual design of a database.

Entity

* An entity can be a real-world object, either animate or inanimate, that can be easily identifiable.
* For instance, within a school database, entities encompass individuals like students and teachers, as well as classes and offered courses.
* All these entities possess specific attributes or properties that define them.

Attributes

* All attributes have values. For example, a student entity may have name, class, and age as attributes.

Relationship

There are 3 types of relationship

* one-to-one
* one-to-many
* many-to-many

Design Guidelines

1. Consider the entities needed to meet the requirements of the website.
2. Start off with one entity that you are sure of. Think about what attributes this entity could have. Then, gradually identify other entities and consider possible attributes for each.
3. If you are stuck at Step 1 and unsure of which entity to create, consider using normalization. This can help you determine the entity by normalizing the attributes to 3NF (Refer to slides in “week 2.1 Introduction to Database Design and Normalization” for normalizations steps).
4. For each entity, determine the necessary constraints. Start by identifying the primary key for each entity. Decide whether the primary key should be auto-incremented. Then, determine which attributes should be unique and which should be set to 'not null'.
5. Once you have identified at least two entities, consider whether there is a relationship between them. If so, what type of relationship is it? If it's a many-to-many relationship, do you need to add an extra table to represent it?
6. After establishing a relationship between two entities, identify where should the foreign key be stored?
7. To test if the design works, add mock data to all the entities you have designed. Perform operations like add, update, and delete to this data. Ensure each operation is successful.
8. Refer to slides in “week 2.0 Introduction to Relational Database” or “week 1.3 – Introduction to Data, Database and Data Models” for ER model design.

# 

# Part A: The Entity-Relationship Diagram

*(Please refer to example of ER diagram in week 2.0 Introduction to Relational Database)*

You will design an Entity-Relationship Model for the conceptual design of your e-commerce website. Create the Entity-Relationship Diagram using the Workbench tool, as covered in the week 2 lesson. Paste a screenshot of your Entity-Relationship Diagram in this section.

Please make your screenshot image are **sharp** and **clearly** shown **all the columns** and **their data type** for each entity.

A diagram of a software company

Description automatically generated with medium confidence

# 

# Part B: The Entity-Attributes Information

In Part A, you designed an Entity-Relationship Diagram. For this section, provide detailed attribute information for **all the entities** present in that diagram.

**For each entity**, obtain the following information:

* Entity Column information
* **All** the **Foreign-Key** information (Some entity will have more than 1 foreign key)

*(Please refer to example in week 2.0 Introduction to Relational Database)*

Order Entity

A screenshot of a computer

Description automatically generated

Foreign key in order table

(order table links to delivery table)

A screenshot of a computer

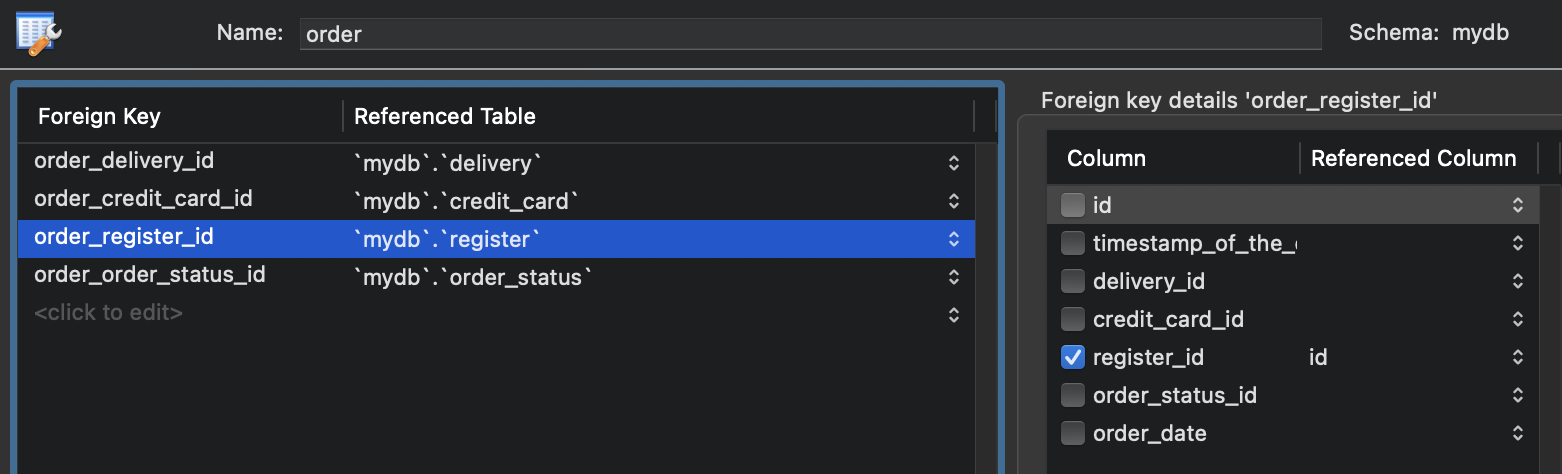
Description automatically generated

(order table links to credit card table)

A screenshot of a computer

Description automatically generated

(order table links to register table)



(order table links to order status table)

A screenshot of a computer

Description automatically generated

**order\_status entity**

**A screenshot of a computer

Description automatically generated**

**foreign key in order\_status entity**

[No foreign key]

**register entity**

A screenshot of a computer

Description automatically generated

**Foreign key in register entity**

[no foreign key]

**login entity**

**A screenshot of a computer

Description automatically generated**

**Foreign key in login entity**

(login table links to register table)

A black and blue line

Description automatically generated with medium confidence

**delivery entity**

**A screenshot of a computer

Description automatically generated**

**Foreign key in delivery table**

[No foreign key]

**credit\_card entity**

**A screenshot of a computer

Description automatically generated**

**Foreign key in credit\_card table**

[No foreign key]

**Order\_item entity**

**A screenshot of a computer

Description automatically generated**

**Foreign key in order\_item table**

(order\_item table links to product\_for\_sale table)

**A screenshot of a computer

Description automatically generated**

(order\_item table links to order table)

A screenshot of a computer

Description automatically generated

**product\_for\_sale entity**

**A screenshot of a computer

Description automatically generated**

**Foreign key in product\_for\_sale table**

(product for sale table links to category table)

**A screenshot of a computer

Description automatically generated**

(product\_for\_sale table links to product\_image table)

A screenshot of a computer

Description automatically generated

**product\_image entity**

**A screenshot of a computer

Description automatically generated**

**Foreign key in product\_image table**

[No foreign key]

**category entity**

**A screenshot of a computer

Description automatically generated**

**Foreign key in category table**

[No foreign key]

# 

# Part C: The Report

Write a report of no more than 400 words to address the following 3 issues:

1)How does your design prevent Insertion anomalies

2)How does your design prevent Update anomalies

3)How does your design prevent Delete anomalies

For each of the 3 points above, elaborate with examples. (Refer to slides in “week 2.1 Introduction to Database Design and Normalization” for anomalies examples).

**Insertion anomalies:**

An insertion anomaly occurs in a database when, due to its structure, one must insert placeholder or unrelated data just to satisfy the requirements for adding a record.

**Example 1:**

**A screenshot of a computer

Description automatically generated**

**Example 2:**

**A screenshot of a computer

Description automatically generated**

**Update anomalies:**

An update anomaly occurs when data is **inconsistently updated** due to the **redundant storage** of the same data in multiple places.

**Example1:**

**A screen shot of a cell phone

Description automatically generated**

**Example 2:**

**A screenshot of a computer

Description automatically generated**

**Delete anomalies:**

A delete anomaly occurs when the deletion of a single piece of data **causes the loss of other related data.**

**Example 1:**

**A screenshot of a computer

Description automatically generated**

**Example 2:**

**A screen shot of a cell phone

Description automatically generated**

**Report**

First example of **insertion anomaly** is product\_for\_sale table and category table. Category table is for the category of the product. If we had category as an attribute inside the product\_for\_sale table that would cause to an insertion anomaly, as this new table would be 1NF. This is because if we wanted to add in a new category but if no product is being sold that can fall within that category, then one must insert placeholder or unrelated data just to satisfy the requirements.

Second example of **insertion anomaly** is order\_table and credit\_card \_table. If credit\_card\_company and the credit\_card fee were within the order table. This would cause an insertion anomaly, and this new table would be 1NF. This is because if wanted to add in a new credit\_card company, for e.g WorldCard but if no one was using WorldCard to pay, then one must insert placeholder or unrelated data just to satisfy the requirements.

The first example of **update anomaly**. Different credit card companies have different transaction fees. If we had made credit card company and the credit card fee an attribute of the order entity it would have caused an **update anomaly**. This is because if a credit card company, e.g MasterCard, decided to increase its credit card payment fees. We would have to change the MasterCard fee not for one record only but for all the order records that are using MasterCard as payment option.

The second example of **update anomaly** are the order table and the delivery table. If delivery\_company and delivery\_fee were attributes within the order table, this would have caused update anomaly. This is because if the deliver company decided to increase its delivery fee. We would have to change all the records of that particular delivery company’s fees.

First example of **delete anomaly** is product\_for\_sale table and category table. For example, if the category was an attribute inside the product\_for\_sale table. And, in the all the products being sold there are only 2 products, Product A & B, which fall under the category wine. If the Product A & B were deleted. In that case, we would lose category (wine) information. However, with category being a separate entity, this insertion anomaly is avoided.

Second example of **delete anomaly** is order table and credit\_card table. For example, if the credit card company and the transaction fee were an attribute of the order table. And if, no one was using the credit card company to pay then we would lose the credit\_card company information as it would not exist in the column, credit\_card company. However, with credit\_card being a separate entity this is avoided.

-End of Report-