**IS 6503 Term Project Deliverable 1 Template**

Your Name:

Your email:

Note that in the first deliverable, each student should independently come up with a business problem/ERD and complete the five sections outlined. Your individual ERD cannot be the same as (or highly similar to) any example covered in the book or on our slides, i.e. your ERD may contain some entities covered in the book examples but **more than 50% of your entities or relationships should be different from any examples/ERDs covered in the book/slides/ assignments.**

1. **Description of the business context and related data management problem(s)**

Perfect Furniture is furniture company that manufactures and sells different kinds of furniture to customers nationwide. The company is currently using different isolated database management systems to store and process data, which causes a lot of inconsistencies and discrepancies across its different business units. It also makes it very difficult for the management to monitor the inventory status and strategically prioritize its production capabilities.

For the Term Project, I would like to create a database application to help Perfect furniture integrate its fragmented data management systems and provide more effective data management…

(This is only an example, please elaborate based on your actual problem/solution)

1. **The entities and the attributes (with description, constraint and data types)**

**Entity**: **Orderline** (Associative Entity between Order and Product)**:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Description:** enter the description of your entity here. For example, the OrderLine entity records each line in an order placed by a customer. | | | |
| **Attribute** | **Description** | **Constraints** | **Data Type** |
| **OrderID** | 6-digit Numeric ID of each order | PK, FK1 | CHAR(12) |
| **ProductID** | 8-digit Numeric ID of the product that appears on each line of the order | PK, FK2 | CHAR(8) |
| OrderedQuantity | Number of units purchased | Unique, Not Null | INTEGER |
| SalesPrice | The final sales price of the product | Not Null, >0 | DECIMAL(4,2) |
| Backordered | Whether this product is backordered | Default value: No | BOOLEAN |
| DateShipped | Date this product is shipped to customer |  | DATE |
| LineMemo | Special notation for each order line |  | VARCHAR2(40) |

…..

Only one sample table provided above. Please use one table for each entity.

1. **ERD (there should be at least 6 entities plus at least two subtypes**, the ERD can be modified in the next deliverable). Your ERD must have all primary/foreign keys, relationships, minim and constraints correctly specified. The ERDs below are just example.

Diagram

Description automatically generated

Below is another example for an ERD that includes the supertype/subtype example for your reference. You should have all entities in one ERD.

Timeline

Description automatically generated

1. **Business rules that define all the relationships among entities, the constraints (if any). Note that each relationship (i.e. each line in your ERD) requires 2 business rules (bi-directional), and each business rule should specify both the minimum and maximum cardinalities.** Use the template below to describe your business rules. Note that you should have at least one of each of the following relationships.

***Relationships:***

**1:1 relationship**: Store – Employee (Manager), each store must have one and only one manager (1,1), whereas each employee may serve as the manger for no store or at most 1 store (0,1) .

**M:N relationship:** Product – Order, a product may appear in no order or many orders (0,M), whereas an order must contain at least 1 and up to many different products (1, M).

**This M:N relationship is broken in 2 1:M relationships** through the associative entity OrderLine which has a 1:M relationship with Product and 1:M relationship with Order.

**1:M relationship**: Customer – Order: a customer can place either no order or multiple orders (0,M), and order can be placed by one and only one customer (1:1).

**1:M Unary relationship**: Employee: An Employee Supervisor may supervise 0 or many employees (0, M), whereas each employee must have one and only one supervisor (1,1)

**M:N Unary relationship (optional extra credit)**: Product Component: A product may serve as the component of 0 or many products (0, M), whereas each product must have 0 or many components (0, M).

**Supertype**: Employee with 3 **subtypes**: Hourly Employee, Salaried Employee, and Consultant, as differentiated by the **subtype discriminator** “Employee Type”

**Disjointed/Overlapping constraint**: The subtypes have a disjointed constraint since an employee can only belong to one of the three subtypes.

**Specialization constraint:** The subtypes have a total specialization constraint since an employee must belong to one of the three subtypes.

Among the above relationships:

**Strong Relationship(s)**: Product – OrderLine, OrderLine – Order.

**Weak Relationship(s)**: Customer – Order.

**Existence Dependency**: Employee - Dependent

**Mandatory relationships** (required a NOT NULL constraint on the FK):

***Constraints***:

1. a product cannot appear in in multiple lines of the same order.

….

1. **A summary of the functionality of the application (i.e. what the users can do with this application, future extension/integration with other systems etc.). These functionalities will be implemented in deliverable 2 and 3 and what you propose can be changed later.**

This new DBMS will allow users to keep track of the number of units available for each product and update the inventory when a sale occurs, it will also allow users to search the database to derive useful business information. For example, a user can search the DBMS to find out which product generates the most sales, which state has the most number of customers, what are the preferences of customers (i.e. the type of furniture ordered) from different states, etc. In the future, we can also connect this DBMS with a shipping tracking system to provide customers more timely update of the shipping status of the products they ordered…..