**School of Computing**

**ST1501 Data Engineering CA1**

**AY2023/2024 Semester 1**

|  |  |
| --- | --- |
| **Adm No** | **P2214296** |
| **Name** | **Ng Wee Herng** |
| **Class** | **DAAA/FT/2A/01** |

|  |  |
| --- | --- |
| **Section A - Database Design (Entity Relationship Diagram)** | 30% |
|  | | |

|  |  |
| --- | --- |
| **Section B - Database Design (Database Diagram)**   * The database supports the described business scenario. * The chosen table names, field names and attributes are descriptive. * The database is normalized. | 10% |
|  | | |

|  |  |
| --- | --- |
| **Section C - Database Creation**   * Attach an SQL file to include create table statememts including the primary key and foreign key definition. * Enter the details of each table in the template. | 10% |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **List all the relations, PK, FK and all non key attributes for the database relation diagram. Attach an SQL file to show the create table SQL script to implement the database design, including the primary key and foreign key definition.** | | | | | |
| **No** | **Relation Name** | **Primary Key** | **Non-key Attribute(s)** | **Foreign Key(s)** | **Referenced Relation** |
| 1 | Store | StoreID | Address | Manager | Employee(Emp\_ID) |
| 2 | Item | Item\_ID | Brand |  |  |
| Description |  |  |
| Price |
| Cost |
| Shape |
| Size |
| UPC |
| Weight |
| Taxable |
| 3 | Customer | Cust\_ID | CustName |  |  |
| Phone |
| Email |
| Date\_Joined |
| 4 | Employee | Emp\_ID | Emp\_Name | StoreRef\_ID | Store(StoreID) |
| SSN |
| Phone |
| Address |
| PayType |
| Password |
| Email | Manager | Employee(Emp\_ID) |
| Date\_Hired |
| Date\_Start |
| Date\_End |
| Pay |
| PassChgDate |
| 5 | Dependents | Emp\_ID | Relationship | Emp\_ID | Employee(Emp\_ID) |
| DependentName | Email |
| Date\_created |
| 6 | Inventory | StoreID | Quantity | StoreID | Store(StoreID) |
| Item\_ID | Item\_ID | Item(Item\_ID) |
| 7 | Checkout | Checkout\_ID | TransactionDate | Emp\_ID | Employee(Emp\_ID) |
| Cust\_ID | Customer(Cust\_ID) |
| Store\_ID | Store(StoreID) |
| 8 | Transactions | Item\_ID | Quantity | Item\_ID | Item(Item\_ID) |
| Checkout\_ID | Checkout\_ID | Checkout(Checkout\_ID) |

|  |
| --- |
| **Section D - The Query Statements 35%** |
| Query 1  Select cus.CustName, co.Store\_ID, s.Address As 'Store Address', e1.Emp\_Name As 'Manager of the Store', co.TransactionDate, i.Brand, i.Description, t.Quantity As 'Quantity Purchased', e2.Emp\_Name As 'Served By'  From Customer cus, Checkout co, Store s, Item i, Transactions t, Employee e1, Employee e2  Where co.Cust\_ID = cus.Cust\_ID  And co.Store\_ID = s.StoreID  And co.Emp\_ID = e2.Emp\_ID  And co.Checkout\_ID = t.Checkout\_ID  And t.Item\_ID = i.Item\_ID  And e1.Emp\_ID = s.Manager |
| Query 1 results |
| Query 2  Select s.Manager As 'Store Manager ID', e.Emp\_Name As 'Name of Manager', s.StoreID as 'Store Managed', i.Item\_ID As Item, i.Quantity As 'Quantity on Inventory'  From Store s, Employee e, Inventory i  Where e.Emp\_ID = s.Manager  And i.StoreID = s.StoreID |
| Query 2 results |
| Query 3  Select Cust\_ID, CustName  From Customer  Where Cust\_ID in  (  Select Cust\_ID  from Checkout  Where Checkout\_ID in  (  Select Checkout\_ID  From Transactions  Group By Checkout\_ID  Having Count(Checkout\_ID) <= 2  )  ) |
| Query 3 results |
| Query 4  Select inv.Item\_ID, i.Description, i.Price \* Sum(inv.Quantity) As Retail, i.Cost \* Sum(inv.Quantity) As Wholesale  from Inventory inv, Item i  Where inv.Item\_ID = i.Item\_ID  Group By inv.Item\_ID, i.Description, i.Price, i.Cost  Having Count(inv.StoreID) >= 2 |
| Query 4 results |
| Query 5  Select e1.Emp\_ID, e1.Emp\_Name, e2.Emp\_ID As 'Manager\_ID', e2.Emp\_Name As 'Manager Name'  From Employee e1, Employee e2  Where e2.Emp\_ID = e1.Manager |
| Query 5 results |
| Query 6  Select e1.Emp\_Name As 'Name of Manager', e1.Manager As 'Boss ID', s.Manager As 'Manager ID', e2.Emp\_Name As 'Name of Boss', s.StoreID, s.Address  From Employee e1, Employee e2, Store s  Where e1.Emp\_ID = s.Manager  And e2.Emp\_ID = e1.Manager  And s.StoreID = e1.StoreRef\_ID  And e1.StoreRef\_ID = e2.StoreRef\_ID |
| Query 6 results |