**Lab 4 : Writing SQL Queries To Create Tables and Populate Them & Writing SQL Queries To Create Views to selectively display only some data or summary data**

**Case Study: Sales Management System (SMS).**

Sales management system database is used to manage sales of products to registered customers. Given below is the data dictionary for SMS.

**Part 1 – To Do:**

**Write SQL Queries to create all the tables as per the configuration stated in the Data Dictionary. After you have created the tables, write queries to populate them with some sample data.**

**Part 2 – To Do:**

**Write SQL Queries to create the SQL Views as per the requirements given in the “Requirements for SQL Views” table.**

**What is a data dictionary ?**

**Data dictionary is a design artifact that defines the structure for each SQL data table.**

**It contains the table name, some details on the purpose of the table, each column’s name, and data type, whether the column is a primary key and/or foreign key, whether the column value is auto generated or user populated, whether the column must have value (null or not null) , whether the column has a default value and if yes what is the default value, the valid range of values that the column can accept.**

**Additionally , you can include the purpose of the column in the data dictionary.**

**DBAs refer to a data dictionary to create a SQL data table and configure integrity rules.**

**In summary, a good data dictionary will have all details needed to create a table. An excellent data dictionary will also include as much integrity rules as possible to ensure the data has high integrity.**

**Built-in integrity rules:-**

1. **primary key**
2. **foreign key**
3. **cascade (optional)**
4. **“not null” setting for columns that must have values.**
5. **range of acceptable values**
6. **computed fields**
7. **usage of lookup tables**

**Data Dictionary for Sales Management System**

Table: Customer

|  |  |  |  |
| --- | --- | --- | --- |
| Purpose: Stores all the registered customer details | | | |
| **Column Name** | **Type** | **Default** | **Note** |
| CustID | Varchar(5) |  | Primary key . Also Serve as their LoginID and DBUserName |
| CustName | Varchar(100) |  | Not null |
| Phone | Varchar(20) |  |  |
| Email | Varchar(200) |  |  |

Table: Product

|  |  |  |  |
| --- | --- | --- | --- |
| Purpose: Stores all the product registered in the system that are available for sales | | | |
| **Column Name** | **Type** | **Default** | **Note** |
| ProductCode | Varchar(10) |  | Primary key |
| ProductName | Varchar(100) |  | Must be unique |
| CostPrice | Decimal(7,2) |  |  |
| SalePrice | Decimal(7,2) |  |  |

Table : Transaction

|  |  |  |  |
| --- | --- | --- | --- |
| Purpose: Records all the high-level transaction details for a purchase | | | |
| Column Name | Type | Default | Note |
| TransactionID | Int |  | Primary key, identity column |
| CustID | Varchar(5) |  | Foreign key references Customer(CustID), not null |
| TransactionDate | DateTime |  |  |

Table: TransactionItem

|  |  |  |  |
| --- | --- | --- | --- |
| Purpose: records all the details of the product sold | | | |
| Column Name | Type | Default | Note |
| TransactionItemID | Int |  | Primary key, identity column |
| TransactionID | Int |  | Foreign key references Transaction (TransactionID) |
| ProductCode | Varchar(10) |  | FK references Product (ProductCode) |
| Quantity | Int | 1 | Must be > 0 |

**Table : Requirements for SQL Views**

|  |  |  |
| --- | --- | --- |
| No | SQL View Name | Purpose |
| 1 | Products\_Limited | To show all columns from Products table except CostPrice. |
| 2 | Purchase\_Summary | Used by customer to check on their purchases. Show a **daily purchase summary** which includes customer name, purchase date and overall total. |

**Sample output for Purchase\_Summary**

SQl Query: **Select \* from Purchase\_Summary**

Output:

A screenshot of a computer

Description automatically generated