

## **A2 (P2) - Case Study – Northwind**

<https://github.com/noct15/infosys222/blob/gh-pages/case/nw.md>

### **Background**

- Northwind is a sample database being commonly used for demos and testing purposes. It contains some business records of a trading company dated from 1996 to 1998
- The database can be created by running a single [SQL script](#)

### **Specification**

- The following specification captures the Northwind database schema through all the CREATE TABLE statements that build up the database:
  - Supplier
  - Category
  - Product
  - Customer
  - Employee
  - Region
  - Territory
  - EmployeeTerritory
  - Shipper
  - Order
  - OrderDetail

### **Supplier**

```
CREATE TABLE Supplier(  
    SupplierID INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,  
    CompanyName TEXT NOT NULL,  
    ContactName TEXT,  
    ContactTitle TEXT,  
    Address TEXT,  
    City TEXT,  
    Region TEXT,  
    PostalCode TEXT,  
    Country TEXT,  
    Phone TEXT,  
    Fax TEXT,  
    HomePage TEXT  
);
```

### **Category**

```
CREATE TABLE Category(  
    CategoryID INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,  
    CategoryName TEXT,  
    Description TEXT  
);
```

## Product

```
CREATE TABLE Product(  
    ProductID INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,  
    ProductName TEXT NOT NULL,  
    SupplierID INTEGER,  
    CategoryID INTEGER,  
    QuantityPerUnit TEXT,  
    UnitPrice REAL DEFAULT 0,  
    UnitsInStock INTEGER DEFAULT 0,  
    UnitsOnOrder INTEGER DEFAULT 0,  
    ReorderLevel INTEGER DEFAULT 0,  
    Discontinued INTEGER NOT NULL DEFAULT 0,  
    CHECK (UnitPrice>=0),  
    CHECK (ReorderLevel>=0),  
    CHECK (UnitsInStock>=0),  
    CHECK (UnitsOnOrder>=0),  
    FOREIGN KEY (CategoryID) REFERENCES Category (CategoryID)  
        ON DELETE NO ACTION ON UPDATE NO ACTION,  
    FOREIGN KEY (SupplierID) REFERENCES Supplier (SupplierID)  
        ON DELETE NO ACTION ON UPDATE NO ACTION  
);
```

## Customer

```
CREATE TABLE Customer  
(  
    CustomerID TEXT NOT NULL,  
    CompanyName TEXT,  
    ContactName TEXT,  
    ContactTitle TEXT,  
    Address TEXT,  
    City TEXT,  
    Region TEXT,  
    PostalCode TEXT,  
    Country TEXT,  
    Phone TEXT,  
    Fax TEXT,  
    PRIMARY KEY (CustomerID)  
);
```

## Employee

```
CREATE TABLE Employee(  
    EmployeeID INTEGER PRIMARY KEY AUTOINCREMENT,  
    LastName TEXT,  
    FirstName TEXT,  
    Title TEXT,  
    TitleOfCourtesy TEXT,  
    BirthDate DATE,  
    HireDate DATE,  
    Address TEXT,  
    City TEXT,  
    Region TEXT,  
    PostalCode TEXT,  
    Country TEXT,  
    HomePhone TEXT,  
    Extension TEXT,  
    Note TEXT,  
    ReportsTo INTEGER,  
    PhotoPath TEXT,  
    FOREIGN KEY (ReportsTo) REFERENCES Employee (EmployeeID)  
        ON DELETE NO ACTION ON UPDATE NO ACTION  
);
```

## Region

```
CREATE TABLE Region(  
    RegionID INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,  
    RegionDescription TEXT NOT NULL  
);
```

## Territory

```
CREATE TABLE Territory(  
    TerritoryID TEXT NOT NULL,  
    TerritoryDescription TEXT NOT NULL,  
    RegionID INTEGER NOT NULL,  
    PRIMARY KEY (TerritoryID),  
    FOREIGN KEY (RegionID) REFERENCES Region (RegionID)  
        ON DELETE NO ACTION ON UPDATE NO ACTION  
);
```

## EmployeeTerritory

```
CREATE TABLE EmployeeTerritory(  
    EmployeeID INTEGER NOT NULL,  
    TerritoryID TEXT NOT NULL,  
    PRIMARY KEY (EmployeeID, TerritoryID),  
    FOREIGN KEY (EmployeeID) REFERENCES Employee (EmployeeID)  
        ON DELETE NO ACTION ON UPDATE NO ACTION,  
    FOREIGN KEY (TerritoryID) REFERENCES Territory (TerritoryID)  
        ON DELETE NO ACTION ON UPDATE NO ACTION  
);
```

## Shipper

```
CREATE TABLE Shipper(  
    ShipperID INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,  
    CompanyName TEXT NOT NULL,  
    Phone TEXT  
);
```

## Order

```
CREATE TABLE 'Order'(  
    OrderID INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,  
    CustomerID TEXT,  
    EmployeeID INTEGER,  
    OrderDate DATETIME,  
    RequiredDate DATETIME,  
    ShippedDate DATETIME,  
    ShipVia INTEGER,  
    Freight NUMERIC DEFAULT 0,  
    ShipName TEXT,  
    ShipAddress TEXT,  
    ShipCity TEXT,  
    ShipRegion TEXT,  
    ShipPostalCode TEXT,  
    ShipCountry TEXT,  
    FOREIGN KEY (EmployeeID) REFERENCES Employee (EmployeeID)  
        ON DELETE NO ACTION ON UPDATE NO ACTION,  
    FOREIGN KEY (CustomerID) REFERENCES Customer (CustomerID)  
        ON DELETE NO ACTION ON UPDATE NO ACTION,  
    FOREIGN KEY (ShipVia) REFERENCES Shipper (ShipperID)  
        ON DELETE NO ACTION ON UPDATE NO ACTION  
);
```

## OrderDetail

```
CREATE TABLE OrderDetail(
    OrderID INTEGER NOT NULL,
    ProductID INTEGER NOT NULL,
    UnitPrice REAL NOT NULL DEFAULT 0,
    Quantity INTEGER NOT NULL DEFAULT 1,
    Discount REAL NOT NULL DEFAULT 0,
    PRIMARY KEY (OrderID, ProductID),
    CHECK (Discount>=0 AND Discount<=1),
    CHECK (Quantity>0),
    CHECK (UnitPrice>=0),
    FOREIGN KEY (OrderID) REFERENCES 'Order' (OrderID)
        ON DELETE NO ACTION ON UPDATE NO ACTION,
    FOREIGN KEY (ProductID) REFERENCES Product (ProductID)
        ON DELETE NO ACTION ON UPDATE NO ACTION
);
```

#### Q01: Who am I (0.2%)

- Use the comment feature of SQL to print your full name, student ID and username in three separate lines at the top of the script file. From this point onwards, you should use the comment feature to separate your answers for each question

#### Q02: What are we selling (0.2%)

- Write a single SQL statement to list all the rows from the Product table. All the columns should be renamed with proper spacing using the alias feature. For example, the UnitPrice column should be renamed as “Unit Price”

#### Q03: Product by price (0.2%)

- Write a single SQL statement to list all the rows from the Product table with ProductName, UnitPrice and UnitsInStock columns only. The rows should be sorted by UnitPrice in descending order

#### Q04: What is the number (0.2%)

- You need to locate the phone number of a shipper company named United Package. Write a single SQL statement to retrieve only that piece of information from the appropriate table. Your SQL statement should cater for different casing scenarios

#### Q05: Fax check (0.2%)

- Write a single SQL statement to list rows from the Customer table only if those customers have a fax number

#### Q06: Show some orders (0.2%)

- Write a single SQL statement to list rows from the Order table when their OrderDate values are within the month of July in 1996

**Q07: Where are you from (0.2%)**

- You want to generate a list of countries that covers all existing customers. Write a single SQL statement to retrieve a country list without any duplication (i.e. the same country should not appear twice in that list) from the Customer table

**Q08: Number of order (0.2%)**

- Write a single SQL statement to return the total number of rows in the Order table. Rename the column in the output as "Number of Order"

**Q09: Two ways to go (0.2%)**

- Write a single SQL statement without using any function to retrieve a list of products where the ProductName is exactly 5 characters long
- Write another single SQL statement that uses appropriate function to do the exact same thing

**Q10: Top 10 products (1%)**

- You are interested to know the top 10 most stocked products in the inventory. Write a single SQL statement to retrieve that information from the Product table with both the ProductName and UnitsInStock columns

**Q11: Tidy list of employee (0.5%)**

- Write a single SQL statement to generate a tidy employee list with only two columns, one with their full name and one with their full address. The full name is composed of the LastName in all caps, the FirstName, and a comma in between

**Q12: Show the details (0.5%)**

- You need to examine the order detail of the order 10250. Write a SQL statement that would retrieve the rows from the OrderDetail table with the columns OrderID, ProductID, UnitPrice, Quantity, Discount and a derived column named Subtotal which is calculated from other columns. For the output, the columns UnitPrice and Subtotal should have a dollar sign (\$) as prefix, and Discount should be shown as a percentage with the sign (%) as suffix

**Q13: Multiple conditions (0.5%)**

- Write a single SQL statement that can list all products with the following features: ProductName begins with C; CategoryID equals to either 1 or 2; UnitPrice that is more than \$20; and Discontinued is false

**Q14: Three new shippers (0.5%)**

- Write the SQL statements to insert 3 rows to the Shipper table

CompanyName	Phone
Trustworthy Delivery	(503) 555-1122
Amazing Pace	(503) 555-3421
Your Named Limited	(503) Your Student ID

- For the last row use your full name as the CompanyName and use the first 7 digits of your own student ID as the number for the Phone after the area code 503. Make sure the ShipperID is automatically generated and so you do not enter them manually

**Q15: How old are you (0.5%)**

- Write a single SQL statement to generate a list of employees with FirstName, LastName and their exact age from the Employee table. For the output, the age should be rounded to the nearest number without any decimal value, and it should be shown as a whole number

**Q16: We are married (0.5%)**

- Two employees of Northwind, Nancy Davolio and Andrew Fuller, are married and you are asked to update Nancy's details in the Employee table. Write a SQL statement to update Nancy's LastName from Davolio to Fuller, and also the TitleOfCourtesy from Ms. to Mrs. Your SQL statement should cater for different casing scenarios

**Q17: We live together (0.5%)**

- Since Nancy has moved in with Andrew after they get married, you need to change her Address, City, Region, PostalCode and HomePhone columns from the Employee table. Write a SQL statement with subqueries to update Nancy's contact details. Your SQL statement should cater for different casing scenarios

**Q18: Product history (0.5%)**

- Write a single SQL statement to create a new table called ProductHistory with the following column specifications:

Column name	Data type	Nullability
ProductID	INTEGER	No
EntryDate	DATE	No
UnitPrice	REAL	Yes
UnitsInStock	INTEGER	Yes
UnitsOnOrder	INTEGER	Yes
ReorderLevel	INTEGER	Yes
Discontinued	INTEGER	No

- You need to set a primary key and a foreign key for the table. Add a primary key constraint based on both the ProductID and EntryDate columns. Add a foreign key constraint based on the ProductID columns of both ProductHistory and Product tables

**Q19: Fill up the history (1%)**

- Write a single SQL statement to fill up the ProductHistory table with all existing rows in the Product table. The EntryDate column should be filled with the current date and time that can be obtained from the appropriate function

**Q20: Good day to hire (1%)**

- Write a single SQL statement to create a list with 2 columns: Day of Week and Hired. The Day of Week column has a possible value ranged from Monday to Sunday, and the Hired column shows the exact number of employees from the Employee table being hired on a particular Day of Week

**Q21: Top sales representative (1%)**

- You are asked to find out the top sales representative (by the accumulated dollar amount of orders) in Northwind. Write a single SQL statement to return the LastName, FirstName, and the total dollar amount of orders associated with that employee under the column Total with a dollar sign (\$) as prefix

**Q22: Who is the boss (2%)**

- Write a single SQL statement with no subquery to create a list with 2 columns: Employee and Manager. The column Employee shows the FirstName of the employee, and the column Manager shows the FirstName of the manager of that employee. Make sure that all employees are listed whether they have a manager or not. For those employees who do not have a manager, the value "No manager" should show up in the second column



### Q23: You have a discount (2%)

- You notice that some products are sold lower than their recommended prices (i.e. the UnitPrice with Discount in OrderDetail table is lower than the UnitPrice in the Product table). Write a single SQL statement to generate a full customer list with five columns:
  - i. CompanyName column from the Customer table renamed as Company;
  - ii. Recommended column with the total amount they should have paid for all the products ordered based on the UnitPrice from the Product table,
  - iii. Ordered column with the total they have actually paid (calculated from the OrderDetail table),
  - iv. Discount column showing discount in absolute value, and
  - v. Percentage column showing discount in percentage value
- All columns with numeric values should be rounded to two decimal places and displayed with appropriate prefix or suffix. Sort the list by the Percentage column in descending order to show which customer is enjoying the highest percentage of discount from Northwind

### Q24: Best shipper (2%)

- According to the Order table, Northwind ships their products to over 20 different countries via three companies in the Shipper table. Over the years staff in Northwind have learnt from which shipper company would be the best for which country. Your task is to reveal that piece of tacit knowledge from the existing data, assuming that the preferred shipper would have earned the highest total Freight from Northwind for shipments to a particular country. Write a single SQL statement with subqueries to generate a list with two columns: the ShipCountry from the Order table and also the preferred CompanyName from the Shipper table for that country