

## Normalization Challenges

### Normalization Challenges

**Challenge 1: Design a database for a Supermarket Billing Application. It should handle Sales, Purchase, Customers, Employees, Suppliers, and Stock maintenance.**

**Possible Fields (Furnish other details as required and you can skip fields as per your design)**

**SuperMarketBilling**(**CustomerId**, CustomerName, CustomerType, CustomerAddress, CustomerMailId, CustomerOrganizationName, CustomerCreditLimit, CustomerAccountNo, **SupplierId**, SupplierName, SupplierType, SupplierPhoneNo, SupplierAddress, SupplierMailId, SupplierOrganizationName, SupplierCreditLimit, SupplierAccountNo, **ProductId**, ProductName, Quantity, UnitPrice, ProductManufactureDate, ProductExpiryDate, **SalesBillNo**, SalesTotalBillAmount, SalesBillProductId, SalesBillProductName, SalesBillProductQuantity, SalesBillAmount, SalesBillDate, SalesBillCustomerId, **PurchaseId**, PurchaseTotalBillAmount, PurchaseBillBatchId, PurchaseDate, **SalesReturnId**, SalesReturnProductId, SalesReturnProductQuantity, SalesReturnReason, SalesReturnAcceptedBy, **EmployeeId**, EmployeeName, EmployeePhone, EmployeeBloodGroup, EmployeeAddress, EmployeeMailId, EmployeeDesignation, EmployeeQualification, EmployeeExperience, **PurchaseReturnId**, PurchaseReturnProductId, PurchaseReturnProductQuantity, PurchaseReturnReason, PurchaseReturnApprovedBy)

### Second Normal Form:

**X- Set of all key attributes.**

{CustomerId, SupplierId, ProductId, SalesBillNo, PurchaseId, SalesReturnId, EmployeeId, PurchaseReturnId}

**All possible subsets of X (set of all key elements) =  $2^n$  where n is the number of elements...**

**Here in this case n is 8 and hence  $2^8$  combinations (i.e. 256 which will be tedious...)**

This is where **Object Oriented Analysis and Design** help us.

For example, **Use Case Diagrams (UML)** will provide us with the required **Separation of Concerns** and hence simplify the normalization process.