**DBMS PROJECT**

**ONLINE INVENTORY MANAGEMENT SYSTEM**

**Key Milestone 2**



## CSE403L Database Management System Lab

Group members

**Arsalan khan (22PWCSE2110)**

**Waseem (22PWCSE2179)**

**Adnan Zeb (22PWCSE2191)**

Class Section: **A**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Student Signature:

Submitted to:

## Engr.Sumayyea Salahuddin

(MAY 27, 2025)

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

## Conversion of Conceptual Schema into Relational Schema

The conceptual schema of the Online Inventory Management Systemincludes several entities, their attributes, and relationships as defined in the entity-relationship diagram (ERD). These have been translated into a relational schema that adheres to the relational database design principles.

**Relational Schema**

1. **Users**

id (PK), name, email, username, password, photo

1. **Suppliers**

id (PK), name, email, phone, address, shopname, type, bank\_name, account\_holder, account\_number, photo

1. **Customers**

id (PK), name, email, phone, address, type, bank\_name, account\_holder, account\_number, photo

1. **Categories**

id (PK), category\_name

1. **Units**

id (PK), unit\_name

1. **Products**

id (PK), product\_name, product\_code, buying\_price, selling\_price, stock, product\_image, category\_id (FK), unit\_id (FK)

1. **Purchases**

id (PK), purchase\_date, purchase\_no, supplier\_id (FK), purchase\_status, created\_by (FK), updated\_by (FK)

1. **purchase\_details**

purchase\_id (PK, FK), product\_id (PK, FK), quantity, unitcost, total

1. **orders**

id (PK), customer\_id (FK), order\_date, order\_status, total\_products, sub\_total, vat, total, invoice\_no, payment\_type, pay\_int, due\_int

1. **order\_details**

order\_id (PK, FK), product\_id (PK, FK), quantity, unitcost, total

## Normalization of Relational Schema to Third Normal Form (3NF)

1. **First Normal Form (1NF)**

Rule: Remove repeating groups — each field must contain only atomic (indivisible) values.

* Your schema already satisfies this
* No multivalued attributes
* Each column holds atomic data (e.g., email, price, quantity)

All tables are in 1NF

1. **Second Normal Form (2NF)**

Rule: 1NF + No partial dependency (non-key attribute must depend on the full primary key)

Applies only to tables with composite primary keys:

* purchase\_details(purchase\_id, product\_id)
* order\_details(order\_id, product\_id)

**In both tables:**

* quantity, unitcost, and total depend on the full key
* No partial dependencies

All other tables have simple primary keys and already satisfy 2NF

1. **Third Normal Form (3NF)**

Rule: 2NF + No transitive dependency (non-key attribute depends only on the primary key)

**products**

* Depends only on id
* category\_id and unit\_id are foreign keys, no derived attributes

**orders**

* All values (e.g., total, VAT, payment) relate directly to the order
* No transitive dependencies

**customers and suppliers**

* All fields directly describe the entity
* No attribute depends on another non-key attribute

**purchase\_details and order\_details**

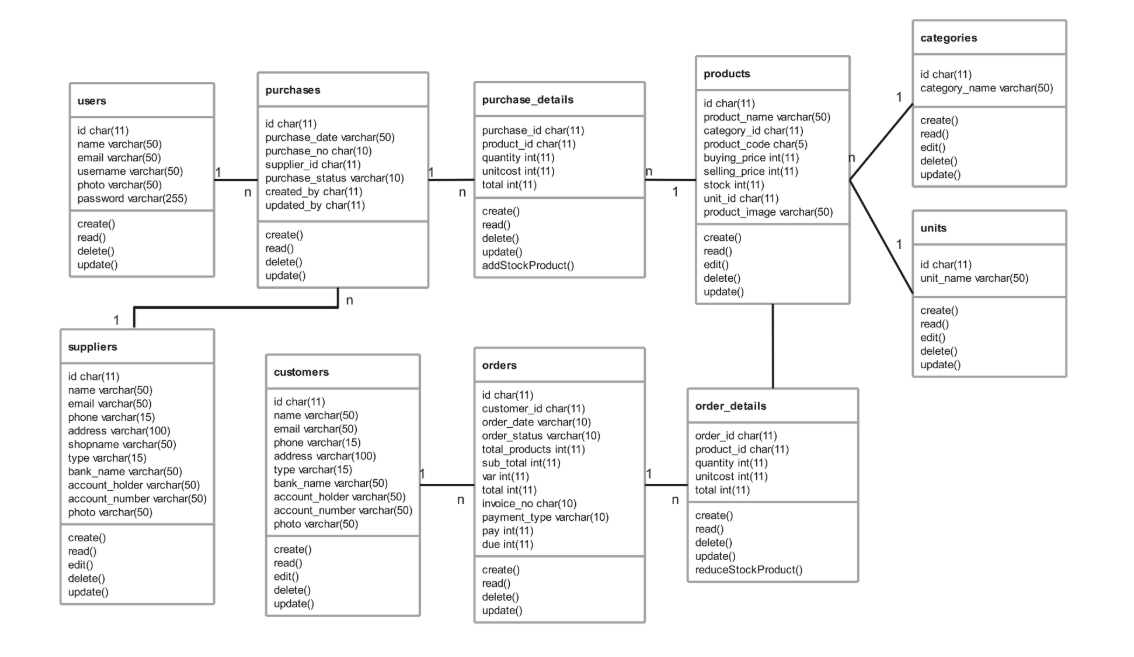
* Composite primary keys
* All fields depend on the combination of keys
* No transitive dependency

**users, categories, units, purchases**

* All attributes depend directly on their primary key
* No transitive dependencies

**Conclusion**

The schema is already in 3NF. Every table:

* Has atomic fields (1NF)
* Has no partial dependencies (2NF)
* Has no transitive dependencies (3NF)

**Refrence:**

<https://chatgpt.com/c/6835c66e-c84c-800a-b82e-64696b01e37a>

https://github.com/fajarghifar/inventory-management-system