

Normalization Process Steps for ERD

Normalization is a database design technique that reduces redundancy and increases data integrity by organizing data into structured tables following normal forms.

To ensure the database schema is normalized, we analyzed its structure based on the normal forms (1NF, 2NF, 3NF, and BCNF). Below is the step-by-step normalization process applied to the schema:

Step 1: First Normal Form (1NF)

Definition of 1NF:

1NF ensures:

- All columns contain atomic values (no repeating or composite fields).
- Each row is uniquely identified using a primary key.
- Each column contains values of a single type (e.g., no lists or arrays in a single column).

Analysis of the ERD in 1NF

- Customers Table:
 - `customer_id` is the primary key, ensuring unique rows.
 - All attributes (`first_name`, `last_name`, `email`, `phone`, `dob`, `gender`) store atomic values.
 - No multi-valued attributes are present.
- Addresses Table:
 - `address_id` is the primary key, uniquely identifying each address.
 - `Customers_customer_id` is a foreign key, linking addresses to customers.
 - Each field (`address_line`, `city`, `state`, `zip_code`, etc.) is atomic.
- Customer_Orders Table:
 - `order_id` is the primary key, uniquely identifying each order.
 - `Customers_customer_id` and `Addresses_address_id` are foreign keys, linking orders to customers and addresses.
 - No multi-valued attributes exist.
- Payments Table:

- payment_id is the primary key.
 - Each attribute (payment_method, status, date, amount_paid) holds atomic values.
- Order_Items Table:
 - order_item_id is the primary key.
 - No composite or multi-valued attributes exist.
 - The table has atomic values (product_quantity, unit_price, product_id, order_id)
- Products Table:
 - product_id is the primary key.
 - category_id and inventory_id are foreign keys, ensuring referential integrity.
 - The table contains atomic values (product_name, price, category_id, inventory_id).
- Categories Table:
 - category_id is the primary key.
 - category_name is a unique column, preventing duplicate categories.
- Discounts Table:
 - discount_id is the primary key.
 - promo_code, discount_percentage, start_date, and end_date are all atomic.
- Returns:
 - return_id is the primary key.
 - Order_Items_order_item_id is a foreign key linking returns to order items.
 - All attributes (return_amount, status, reason, returned_quantity) are atomic.
- Supplier_Products:
 - id is the primary key.
 - Products_product_id and Suppliers_supplier_id are foreign keys.
 - The table contains atomic values (supply_price, product_id, supplier_id).
- Suppliers:
 - supplier_id is the primary key.
 - supplier_name, contact_number, email are all atomic values.
- Inventory:
 - inventory_id is the primary key.
 - Warehouses_warehouse_id is a foreign key.
 - The table contains atomic values (stock_level, last_restock_date, reorder_threshold).
- Warehouse_Orders:
 - order_id is the primary key.

- Warehouses_warehouse_id, Suppliers_supplier_id, and Inventory_inventory_id are foreign keys.
- The table contains atomic values (order_date, total_quantity, warehouse_id, supplier_id, inventory_id).
- Warehouses:
 - warehouse_id is the primary key.
 - The table contains atomic values (warehouse_code, city, state, country, manager_name, contact_number).

The database satisfies 1NF because all fields contain atomic values, and each table has a primary key.

Step 2: Second Normal Form (2NF)

Definition of 2NF:

- The database must be in 1NF.
- No partial dependency should exist, meaning every non-key column should be fully functionally dependent on the entire primary key.

Identifying Partial Dependencies

- If a table has a composite primary key, its non-key attributes must depend on both parts of the key.

Analysis of the ERD in 2NF

- Supplier_Products Table:
 - supply_price depends on both product_id and supplier_id.
- Customer_Orders Table:
 - The primary key is order_id.
 - total_amount, order_date, and status fully depend on order_id, so no partial dependency exists.
- Order_Items Table:
 - The primary key is order_item_id, ensuring each row is uniquely identified.
 - product_quantity, unit_price, Products_product_id, and Customer_Orders_order_id fully depend on order_item_id.

- Payments Table:
 - The primary key is payment_id, and payment_method, status, date, and amount_paid fully depend on it.
- Warehouse_Orders (order_id, warehouse_id, supplier_id, inventory_id)
 - total_quantity depends on all keys.
 - No partial dependencies.

All tables satisfy 2NF because there are no partial dependencies.

Step 3: Third Normal Form (3NF)

Definition of 3NF:

- The database must be in 2NF.
- There should be no transitive dependency, meaning no non-key attribute should depend on another non-key attribute.

Analysis of the ERD in 3NF

- Customers Table:
 - customer_id → first_name, last_name, email, phone, dob, gender
 - (No transitive dependency)
- Addresses Table:
 - address_id → address_line, city, state, zip_code, Customers_customer_id
 - (No transitive dependency)
- Customer_Orders Table:
 - order_id → order_date, total_amount, status, Customers_customer_id, Addresses_address_id
 - (No transitive dependency)
- Products Table:
 - product_id → product_name, price, Categories_category_id, Inventory_inventory_id
 - Categories_category_id is a foreign key and does not introduce redundancy.
- Suppliers Table:
 - supplier_id → supplier_name, contact_number, email
 - (No transitive dependency)
- Categories Table:

- category_id → category_name
- (No transitive dependency)

The schema satisfies 3NF because there are no transitive dependencies.

Step 4: Boyce-Codd Normal Form (BCNF)

Definition of BCNF:

- The database must be in 3NF.
- Every determinant must be a candidate key (i.e., any column that uniquely determines another column must itself be a candidate key).

Analysis of the ERD in BCNF

- Customers Table:
 - customer_id is the only determinant
- Addresses Table:
 - address_id is the only determinant
- Customer_Orders Table:
 - order_id is the only determinant
- Payments Table:
 - payment_id is the only determinant
- Products Table:
 - product_id is the only determinant
- Supplier_Products Table:
 - The primary key is (id, product_id, supplier_id).
 - No other column acts as a determinant.
- Categories Table:
 - category_id is the primary key.
 - category_name is marked as a unique constraint, ensuring no duplicate categories.

All tables are in BCNF as every determinant is a candidate key.

Conclusion

The given ERD is fully normalized up to BCNF, ensuring:

- Minimal redundancy
- No insertion, update, or deletion anomalies
- Optimal data integrity and consistency

This normalization process makes the database efficient and well-structured for handling real-world applications.