Relational Schema for E-Commerce Database

Entities and Attributes

customers (<u>customer_id</u>, user_id, password)

- Primary Key: customer_id
- Constraints: user_id NOT NULL, password NOT NULL

customer_accounts (<u>customer_account_id</u>, bank_account_number, bank_name, customer_id)

- Primary Key: customer_account_id
- Foreign Key: customer_id REFERENCES customers(customer_id)
- Constraints: bank_account_number NOT NULL, bank_name NOT NULL

customer_orders (<u>customer_order_id</u>, customer_id, total_price, customer_account_id)

- Primary Key: customer_order_id
- Foreign Keys:
  - customer_id REFERENCES customers(customer_id)
  - customer_account_id REFERENCES customer_accounts(customer_account_id)
- Constraints: total_price NOT NULL

items (<u>item_id</u>, name)

- Primary Key: item_id
- Constraints: name NOT NULL

class (<u>class_id</u>, item_id, max_price, class_type)

- Primary Key: class_id
- Foreign Key: item_id REFERENCES items(item_id)
- Constraints:
  - max_price NOT NULL
  - class_type ENUM('A', 'B', 'C')

suppliers (<u>supplier_id</u>, name)

- Primary Key: supplier_id
- Constraints: name NOT NULL

order_items (<u>order_item_id</u>, customer_order_id, quantity, price, supplier_id, class_id, item_id)

- Primary Key: order_item_id
- Foreign Keys:
  - customer_order_id REFERENCES customer_orders(customer_order_id)
  - supplier_id REFERENCES suppliers(supplier_id)
  - class_id REFERENCES class(class_id)
  - item_id REFERENCES items(item_id)
- Constraints: quantity NOT NULL, price NOT NULL

supplier_items (<u>supplier_item_id</u>, supplier_id, commission, class_id, discount, stock, review, number_reviews, item_id)

- Primary Key: supplier_item_id
- Foreign Keys:
  - supplier_id REFERENCES suppliers(supplier_id)
  - class_id REFERENCES class(class_id)
  - item_id REFERENCES items(item_id)
- Constraints:
  - commission NOT NULL
  - stock DEFAULT 0
  - number_reviews DEFAULT 0

preference_index (<u>preference_index_id</u>, supplier_item_id, index_val)

- Primary Key: preference_index_id
- Foreign Key: supplier_item_id REFERENCES supplier_items(supplier_item_id)

Functional Dependencies

customers

- customer_id → user_id, password

customer_accounts

- customer_account_id → bank_account_number, bank_name, customer_id
- (customer_id, bank_account_number) → customer_account_id, bank_name

customer_orders

- customer_order_id → customer_id, total_price, customer_account_id

items

- item_id → name

class

- class_id → item_id, max_price, class_type
- (item_id, class_type) → class_id, max_price

suppliers

- supplier_id → name

order_items

- order_item_id → customer_order_id, quantity, price, supplier_id, class_id, item_id

supplier_items

- supplier_item_id → supplier_id, commission, class_id, discount, stock, review, number_reviews, item_id
- (supplier_id, item_id, class_id) → supplier_item_id, commission, discount, stock, review, number_reviews

preference_index

- preference_index_id → supplier_item_id, index_val
- supplier_item_id → index_val

Specialization/Generalization Relationships

1. **Item Classification**:
   - Each item can be classified into different classes (A, B, C)
   - The class table represents a specialization of items with specific pricing and type attributes
2. **Item Supply**:
   - Items are supplied by different suppliers with varying terms (commission, discount, stock)
   - The supplier_items table represents this many-to-many relationship with additional attributes

Business Rules and Constraints

1. The preference_index is calculated using the function `Calculate_preference_index(commission, review, discount)` with the formula: `1 / (1 + exp((-1 * 1.0 * commission) + (-1 * 1.0 * review) + (-1`

`* discount)))`

2. When a supplier item's attributes (commission, review, discount) are updated, a trigger automatically recalculates the preference index
3. Each item must be associated with a class (A, B, or C) with specific pricing constraints
4. Customer orders are linked to a specific customer account for payment
5. Order items connect customer orders with specific items, suppliers, and item classes

Normal Form Analysis

The schema is in Third Normal Form (3NF) because:

1. It is in 1NF: All attributes contain atomic values and there are no repeating groups.
2. It is in 2NF: All non-key attributes are fully functionally dependent on their respective primary keys.
3. It is in 3NF: There are no transitive dependencies; all non-key attributes depend only on the primary key.

Some tables (like order_items and supplier_items) contain multiple foreign keys, but this is appropriate for their role as association tables in a many-to-many relationship.