



CS2102 - Database System

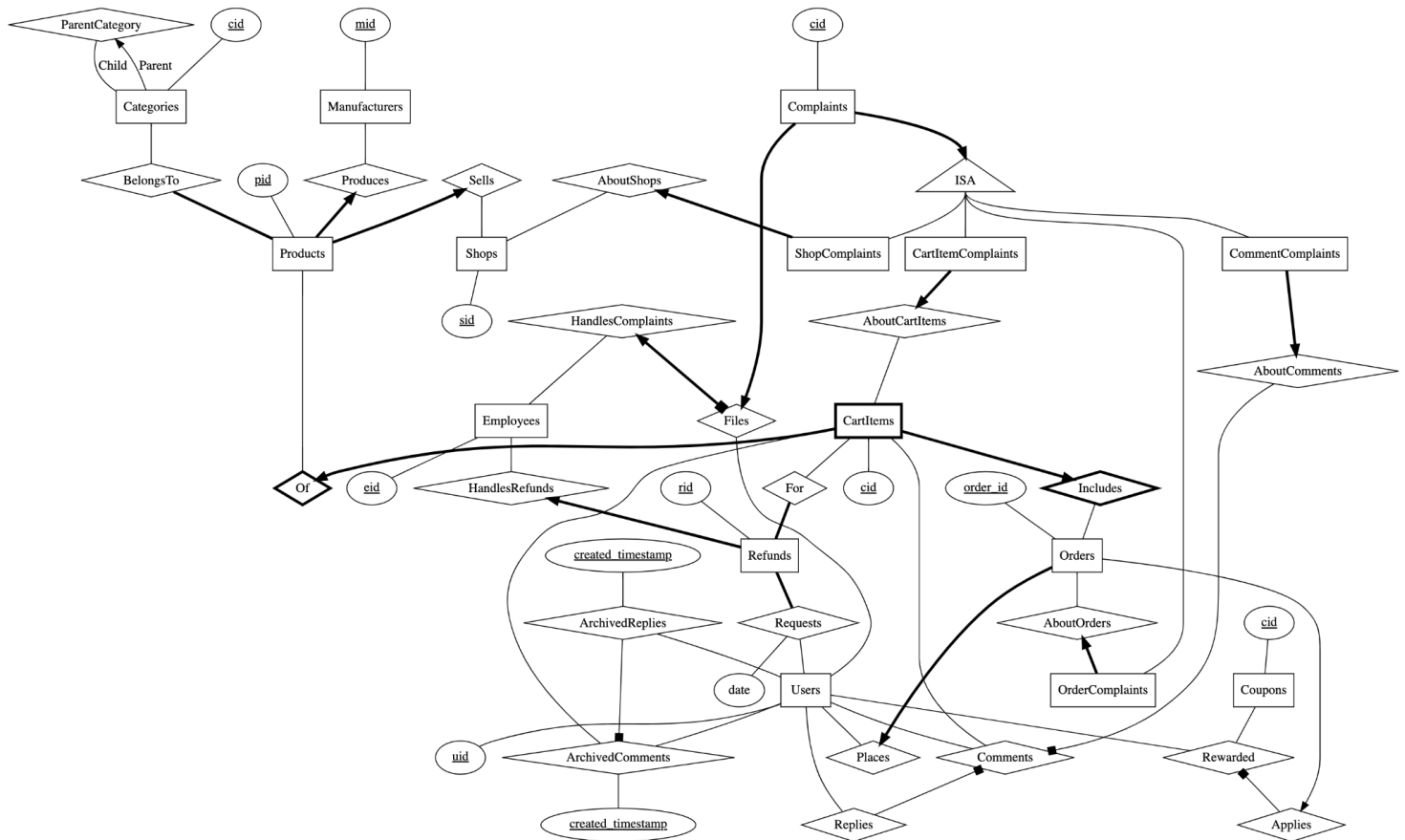
Project (Part 1) Report

Team 36

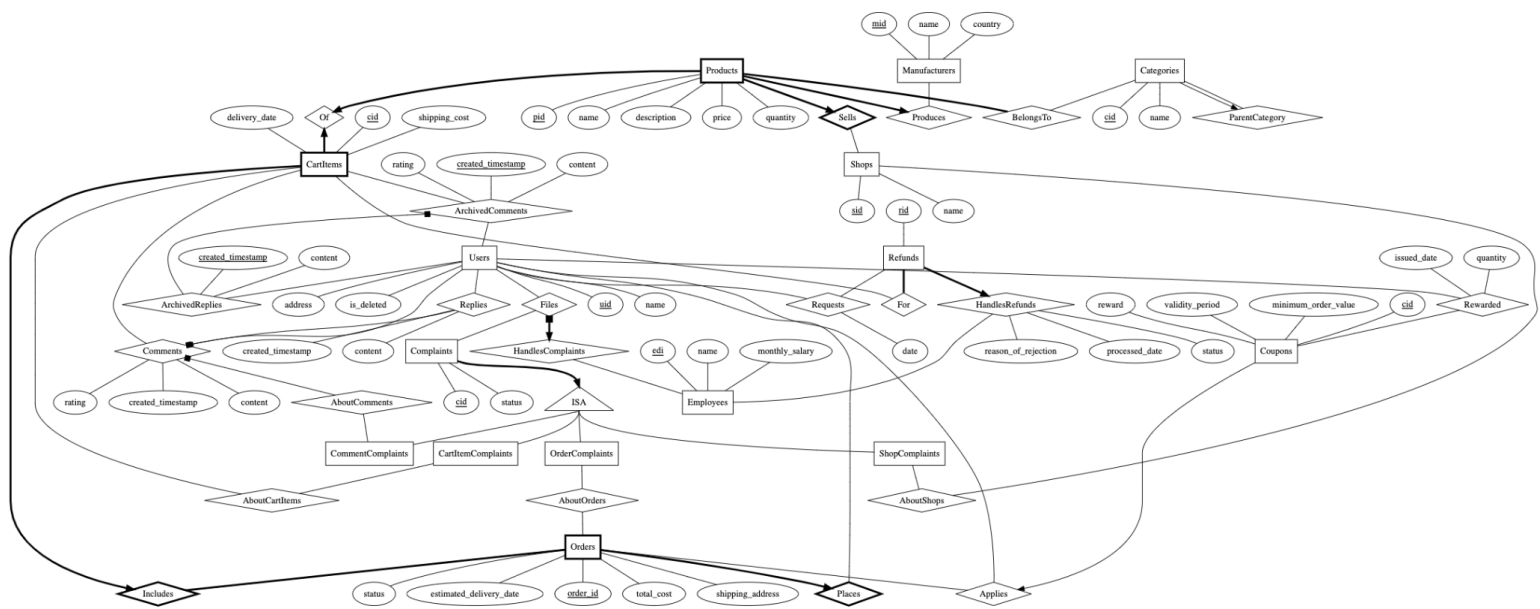
Member	NUS ID
Nguyen Van Binh	e0550453
Pham Ba Thang	e0550373
Le Ngoc Quynh Mai	e0550399
Le Nguyen Quang Dang Khoa	e0550384

1. ER Model of the application

Below is the simplified ER model (with non-key attributes omitted) of the Adazal application.



The full ER model is attached to the next page.



2. The relational database schema

```
DROP TABLE IF EXISTS
    Shops, Manufacturers, Categories, Products, Employees, Users, Orders,
    CartItems, Refunds, HandlesRefunds, Comments, ArchivedComments, Replies,
    ArchivedReplies, Coupons, Rewarded, Applies, Complaints, CartItemComplaints,
    ShopComplaints, OrderComplaints, CommentComplaints, Files, HandlesComplaints
CASCADE;

CREATE TABLE Shops (
    sid INTEGER PRIMARY KEY,
    name VARCHAR(128)
);

CREATE TABLE Manufacturers (
    mid INTEGER PRIMARY KEY,
    name VARCHAR(128),
    country VARCHAR(128)
);

CREATE TABLE Categories (
    cid INTEGER PRIMARY KEY,
    name VARCHAR(128),
    parent_id INTEGER DEFAULT NULL REFERENCES Categories,
    CHECK (cid IS DISTINCT FROM parent_id)
);

CREATE TABLE Products (
    pid INTEGER,
    name VARCHAR(128),
    category_id INTEGER NOT NULL REFERENCES Categories ON UPDATE CASCADE,
    manufacturer_id INTEGER NOT NULL REFERENCES Manufacturers ON UPDATE CASCADE,
    shop_id INTEGER NOT NULL REFERENCES Shops ON DELETE CASCADE ON UPDATE CASCADE,
    description TEXT,
    price NUMERIC CHECK (price >= 0),
    quantity INTEGER CHECK (quantity >= 0),
    PRIMARY KEY (pid)
);

CREATE TABLE Employees (
    eid INTEGER PRIMARY KEY,
    name TEXT,
    monthly_salary NUMERIC CHECK (monthly_salary >= 0)
);

CREATE TABLE Users (
    uid INTEGER PRIMARY KEY,
    name TEXT,
```

```

    address TEXT,
    is_deleted BOOLEAN DEFAULT FALSE
);

CREATE TABLE Orders (
    order_id INTEGER,
    user_id INTEGER NOT NULL REFERENCES Users ON UPDATE CASCADE ON DELETE CASCADE,
    total_cost NUMERIC CHECK (total_cost >= 0),
    shipping_address TEXT,
    PRIMARY KEY (order_id),
    UNIQUE (order_id, user_id)
);

CREATE TABLE CartItems (
    cid INTEGER PRIMARY KEY,
    product_id INTEGER NOT NULL,
    order_id INTEGER NOT NULL,
    user_id INTEGER NOT NULL,
    status TEXT CHECK (status IN ('being processed', 'shipped', 'delivered')),
    quantity INTEGER CHECK (quantity > 0),
    shipping_cost NUMERIC CHECK (shipping_cost >= 0),
    estimated_delivery_date DATE,
    delivery_date DATE,
    FOREIGN KEY (product_id) REFERENCES Products ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (order_id, user_id) REFERENCES Orders (order_id, user_id)
        ON DELETE CASCADE ON UPDATE CASCADE,
    UNIQUE (order_id, product_id),
    UNIQUE (order_id, product_id, user_id)
);

CREATE TABLE Refunds (
    rid INTEGER,
    user_id INTEGER REFERENCES Users ON UPDATE CASCADE,
    quantity INTEGER CHECK (quantity > 0),
    product_id INTEGER NOT NULL,
    order_id INTEGER NOT NULL,
    date DATE,
    FOREIGN KEY (product_id, order_id) REFERENCES CartItems (product_id, order_id)
        ON DELETE CASCADE ON UPDATE CASCADE,
    PRIMARY KEY (rid)
);

CREATE TABLE HandlesRefunds (
    refund_id INTEGER REFERENCES Refunds ON UPDATE CASCADE ON DELETE CASCADE,
    employee_id INTEGER REFERENCES Employees ON UPDATE CASCADE ON DELETE CASCADE,
    status TEXT CHECK (status IN ('processing', 'accepted', 'rejected')),
    reason_of_rejection TEXT,
    processed_date DATE,
    PRIMARY KEY (refund_id),
    CHECK ((reason_of_rejection IS NULL) OR (status = 'rejected'))
);

```

```

CREATE TABLE Comments (
    user_id INTEGER NOT NULL,
    order_id INTEGER NOT NULL,
    product_id INTEGER NOT NULL,
    created_timestamp TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
    content TEXT,
    rating INTEGER CHECK (rating IN (1,2,3,4,5)),
    FOREIGN KEY (user_id, order_id, product_id) REFERENCES CartItems (user_id, order_id, product_id)
        ON DELETE CASCADE ON UPDATE CASCADE,
    PRIMARY KEY (user_id, product_id),
    CHECK (NOT ((content IS NULL) AND (rating IS NULL)))
);

CREATE TABLE ArchivedComments (
    user_id INTEGER NOT NULL,
    order_id INTEGER NOT NULL,
    product_id INTEGER NOT NULL,
    created_timestamp TIMESTAMP WITH TIME ZONE,
    content TEXT,
    rating INTEGER CHECK (rating IN (1,2,3,4,5)),
    FOREIGN KEY (user_id, order_id, product_id) REFERENCES CartItems (user_id, order_id, product_id)
        ON UPDATE CASCADE,
    PRIMARY KEY (user_id, product_id, created_timestamp)
);

CREATE TABLE Replies (
    commenter_id INTEGER NOT NULL,
    product_id INTEGER NOT NULL,
    replier_id INTEGER REFERENCES Users ON UPDATE CASCADE ON DELETE CASCADE,
    content TEXT,
    created_timestamp TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (commenter_id, product_id) REFERENCES Comments (user_id, product_id)
        ON DELETE CASCADE ON UPDATE CASCADE,
    PRIMARY KEY (commenter_id, product_id, replier_id)
);

CREATE TABLE ArchivedReplies (
    commenter_id INTEGER NOT NULL,
    product_id INTEGER NOT NULL,
    comment_timestamp TIMESTAMP WITH TIME ZONE,
    replier_id INTEGER REFERENCES Users ON UPDATE CASCADE ON DELETE CASCADE,
    content TEXT,
    created_timestamp TIMESTAMP WITH TIME ZONE,
    FOREIGN KEY (commenter_id, product_id, comment_timestamp) REFERENCES ArchivedComments (user_id,
product_id, created_timestamp) ON UPDATE CASCADE,
    PRIMARY KEY (commenter_id, product_id, replier_id, created_timestamp)
);

CREATE TABLE Coupons (
    cid INTEGER PRIMARY KEY,

```

```

    reward NUMERIC CHECK (reward >= 0),
    validity_period INTERVAL,
    minimun_order_value NUMERIC CHECK (minimun_order_value >= 0)
);

CREATE TABLE Rewarded (
    user_id INTEGER REFERENCES Users ON UPDATE CASCADE,
    coupon_id INTEGER REFERENCES Coupons ON UPDATE CASCADE ON DELETE CASCADE,
    issued_date DATE,
    quantity INTEGER CHECK (quantity >= 0),
    PRIMARY KEY (user_id, coupon_id)
);

CREATE TABLE Applies (
    user_id INTEGER,
    order_id INTEGER REFERENCES Orders ON UPDATE CASCADE ON DELETE CASCADE,
    coupon_id INTEGER,
    FOREIGN KEY (user_id, coupon_id) REFERENCES Rewarded (user_id, coupon_id)
        ON UPDATE CASCADE ON DELETE CASCADE,
    PRIMARY KEY (order_id)
);

CREATE TABLE Complaints (
    cid INTEGER PRIMARY KEY,
    status TEXT CHECK (status IN ('pending', 'being processed', 'addressed'))
);

CREATE TABLE CartItemComplaints (
    cid INTEGER PRIMARY KEY REFERENCES Complaints ON DELETE CASCADE ON UPDATE CASCADE,
    order_id INTEGER NOT NULL,
    product_id INTEGER NOT NULL,
    FOREIGN KEY (order_id, product_id) REFERENCES CartItems (order_id, product_id) ON DELETE CASCADE
ON UPDATE CASCADE
);

CREATE TABLE ShopComplaints (
    cid INTEGER PRIMARY KEY REFERENCES Complaints ON DELETE CASCADE ON UPDATE CASCADE,
    shop_id INTEGER REFERENCES Shops ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE OrderComplaints (
    cid INTEGER PRIMARY KEY REFERENCES Complaints ON DELETE CASCADE ON UPDATE CASCADE,
    order_id INTEGER REFERENCES Orders ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE CommentComplaints (
    cid INTEGER PRIMARY KEY REFERENCES Complaints ON DELETE CASCADE ON UPDATE CASCADE,
    user_id INTEGER,
    product_id INTEGER,
    FOREIGN KEY (user_id, product_id) REFERENCES Comments ON DELETE CASCADE ON UPDATE CASCADE
);

```

```
);

CREATE TABLE Files (
    user_id INTEGER REFERENCES Users ON UPDATE CASCADE ON DELETE CASCADE,
    complaint_id INTEGER REFERENCES Complaints ON DELETE CASCADE ON UPDATE CASCADE,
    PRIMARY KEY (complaint_id)
);

CREATE TABLE HandlesComplaints (
    complaint_id INTEGER,
    FOREIGN KEY (complaint_id) REFERENCES Files (complaint_id)
        ON DELETE CASCADE ON UPDATE CASCADE,
    employee_id INTEGER REFERENCES Employees ON DELETE CASCADE ON UPDATE CASCADE,
    PRIMARY KEY (complaint_id, employee_id)
);
```

3. Justification for any non-trivial design decisions made

- User can request multiple Refunds for a CartItems (No constraint stated in problem description), therefore, normal relationship is used instead of key constraint from CartItems to Refunds in ER diagram.
- VARCHAR(128) is used instead of TEXT as the type for the names of some entities as well as the country attribute of Manufacturers since those attributes should not be too long and doing so may benefit the speed of the system.
- The CartItems entity is used to model the items of 1 product type in Orders, and an Order made by a User is composed of many CartItems. The status, quantity, shipping cost, product type, delivery date, and estimated delivery date is captured within this entity.
- Refunds entity is connected to CartItems through relation For, while Coupon entity is connected to Order through relation Applies. There is no connection between For and Applies, so the constraint “the effects of coupons are ignored during product refunds” is enforced.
- After an Employee handles a Refund, the Refund may be accepted or rejected. We use the relation HandlesRefunds to model this activity. If the status of HandlesRefunds is ‘rejected’, the reason_of_rejection must be specified, otherwise (status is either ‘processing’ or ‘accepted’) it has to be null.
- ArchivedComments and ArchivedReplies are created the same as Comments and Replies with two changes, which are ON DELETE CASCADE is not included in the foreign key and created_timestamp being in the primary key. This is to make sure everything is stored even when other tables are deleted and if users modify comments or replies, the previous comment are still stored in the table.
- Applies references to Rewarded so that we can enforce Users to can only apply the coupons that are rewarded to them.
- is_deleted is a boolean attribute of Users with the default value false. When the user deletes his/her account, the corresponding User will not be deleted in the database, instead, its is_deleted attribute will be set to true, and in the UI, the displayed name of that user will be “A Deleted User” while all the information of the user is not deleted.

4. Application's constraints that are not captured by the proposed ER data model

- Refund requests can only be made within 30 days from the delivery date of a product
- Refund quantity \leq CartItems quantity
- CartItems quantity \leq Shops quantity of the product
- The status of complaint is only changed from "pending" to "being processed" after it is picked up by an Adazal employee.

5. Application's constraints that are not captured by the proposed relational database schema

- Refund request can only be made within 30 days from the delivery date of a product
- Refund quantity \leq CartItems quantity
- CartItems quantity \leq Shops quantity of the product
- When the status of complaint is "pending" the Adazal employee must be NULL (because the status of complaint is only changed from "pending" to "being processed" after it is picked up by an Adazal employee)