Kharbash Mohammed KH H Al-Mansoori

Homework 3

Part 1: Real Estate Company Database Design

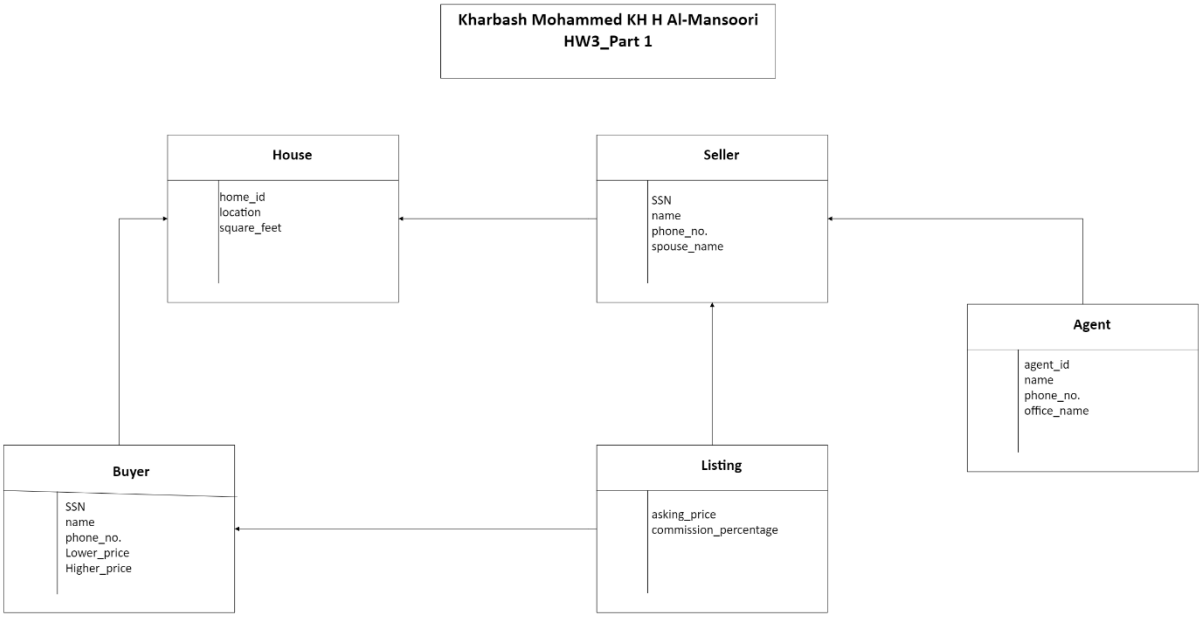

Figure 1: Real Estate Company Database Design

Entities and Attributes

This database represents a real estate company's operations and includes four key entities: House, Seller, Agent, and Buyer. Each entity has associated attributes:

1. **House**:
    - home_id (Primary Key): Unique ID for each house.
    - location: Composed of street address and state name.
    - square_feet: The total area of the house.
2. **Seller**:
    - SSN (Primary Key): Social Security no. to identify the seller.
    - name: The seller's name.

    - phone_no.: Seller's contact no.
    - spouse_name: Optional, name of the spouse.
3. **Agent**:
    - agent_id (Primary Key): Unique identifier for each agent.
    - name: The agent's name.
    - phone_no.s: One or more contact no.s for the agent.
    - office_name: Name of the agent's office.
4. **Buyer**:
    - SSN (Primary Key): Social Security no. for identifying the buyer.
    - name: The buyer's name.
    - phone_no.: Contact no. for the buyer.
    - price_range_lower and price_range_upper: The buyer's expected price range.

Relationships

1. **House - Seller**: One seller owns each house, but a seller may own multiple houses.
2. **Seller - Agent**: A seller can list houses through an agent, and one agent can list multiple houses for various sellers.
3. **House - Agent**: Only one agent lists a house at a time, though agents may list many houses.
4. **Agent - Buyer**: Each agent represents multiple buyers, but a buyer is associated with only one agent.

Additional Assumptions

- A separate "Listing" entity could track the association between a house and its seller, including price and commission details.

- Houses must have sellers before being listed, and only one agent can list a house at a time.
- Buyer data is stored only when associated with an agent.

Part 2: Snack Distributor Database Design

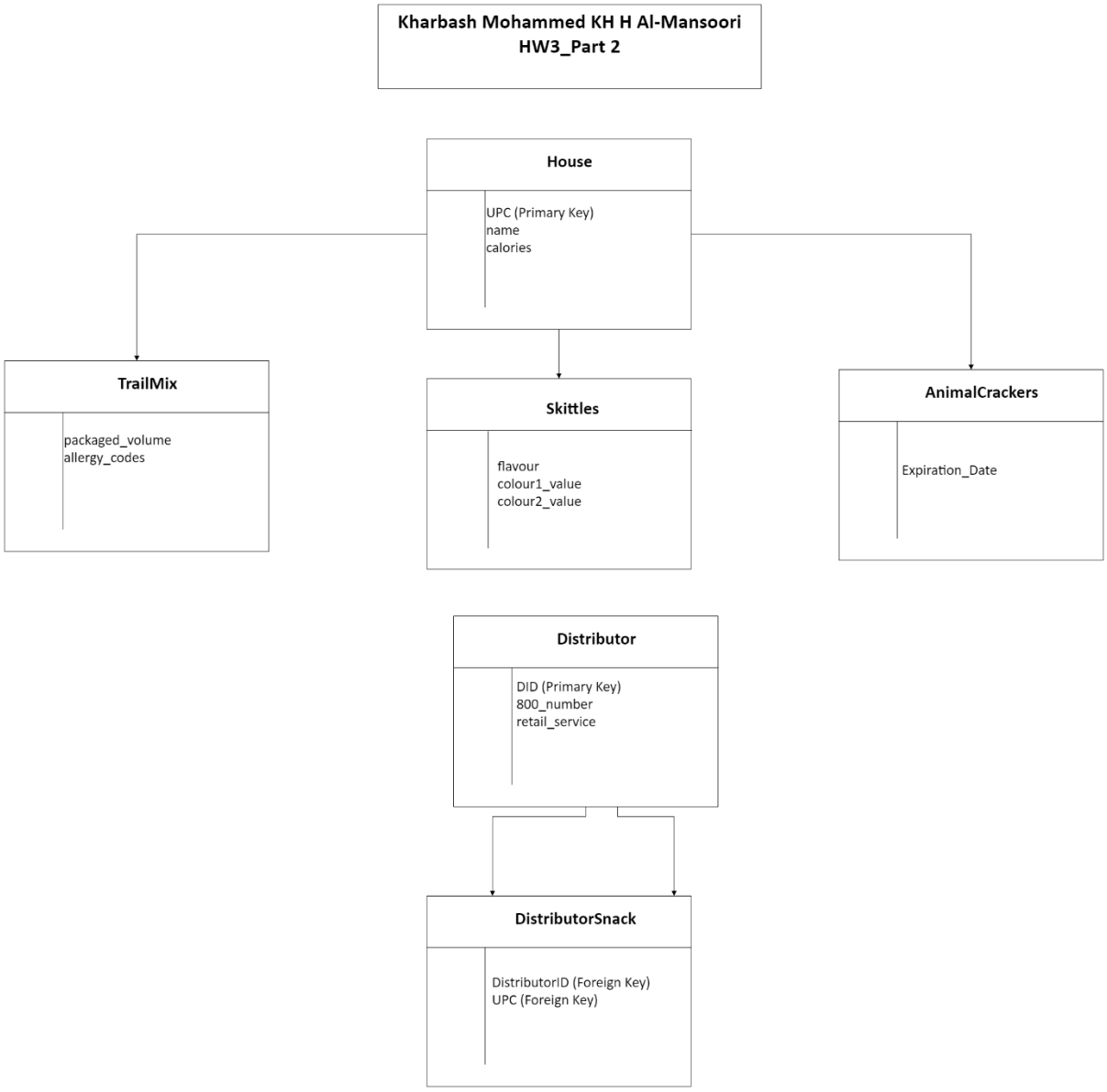

Figure 2: Snack Distributor Database Design

Entities and Attributes

The snack distributor's database uses a superclass/subclass model, representing different snack types and distributors. The "Snack" entity is a superclass, while "Skittles," "Animal Crackers," and "Trail Mix" are subclasses.

1. **Snack (Superclass)**:
    - `UPC` (Primary Key): Universal Product Code for identifying snacks.
    - `name`: The snack's name.
    - `calories`: Total calorie count per package.
2. **Skittles (Subclass)**:
    - `UPC` (Foreign Key): Linked to the Snack entity.
    - `flavor`: The flavor of the Skittles.
    - `color_1_value`, `color_2_value`: Colors that appear in the package.
3. **Animal Crackers (Subclass)**:
    - `UPC` (Foreign Key): Linked to the Snack entity.
    - `expiration_date`: The snack's expiration date.
4. **Trail Mix (Subclass)**:
    - `UPC` (Foreign Key): Linked to the Snack entity.
    - `packaged_volume`: Estimated package volume.
    - `allergy_codes`: Any relevant allergy warnings.
5. **Distributor**:
    - `DID` (Primary Key): Unique ID for each distributor.
    - `800_no.`: Contact no. for customer service.
    - `retail_service`: Indicates if the distributor offers retail services.
6. **Distributor_Snack (Relationship)**:
    - `Distributor_ID` (Foreign Key): Links to the Distributor entity.

- `UPC` (Foreign Key): Links to the Snack entity.

Relationships

- Each snack is associated with a unique record in the Snack entity.
- Distributors can supply multiple snacks, and a snack may be offered by multiple distributors.

Assumptions

- UPC codes are unique to ensure there are no duplicate snacks.
- The subclasses (Skittles, Animal Crackers, and Trail Mix) have one-to-one relationships with the Snack entity.
- The "Distributor_Snack" entity enables a many-to-many relationship between distributors and snacks.