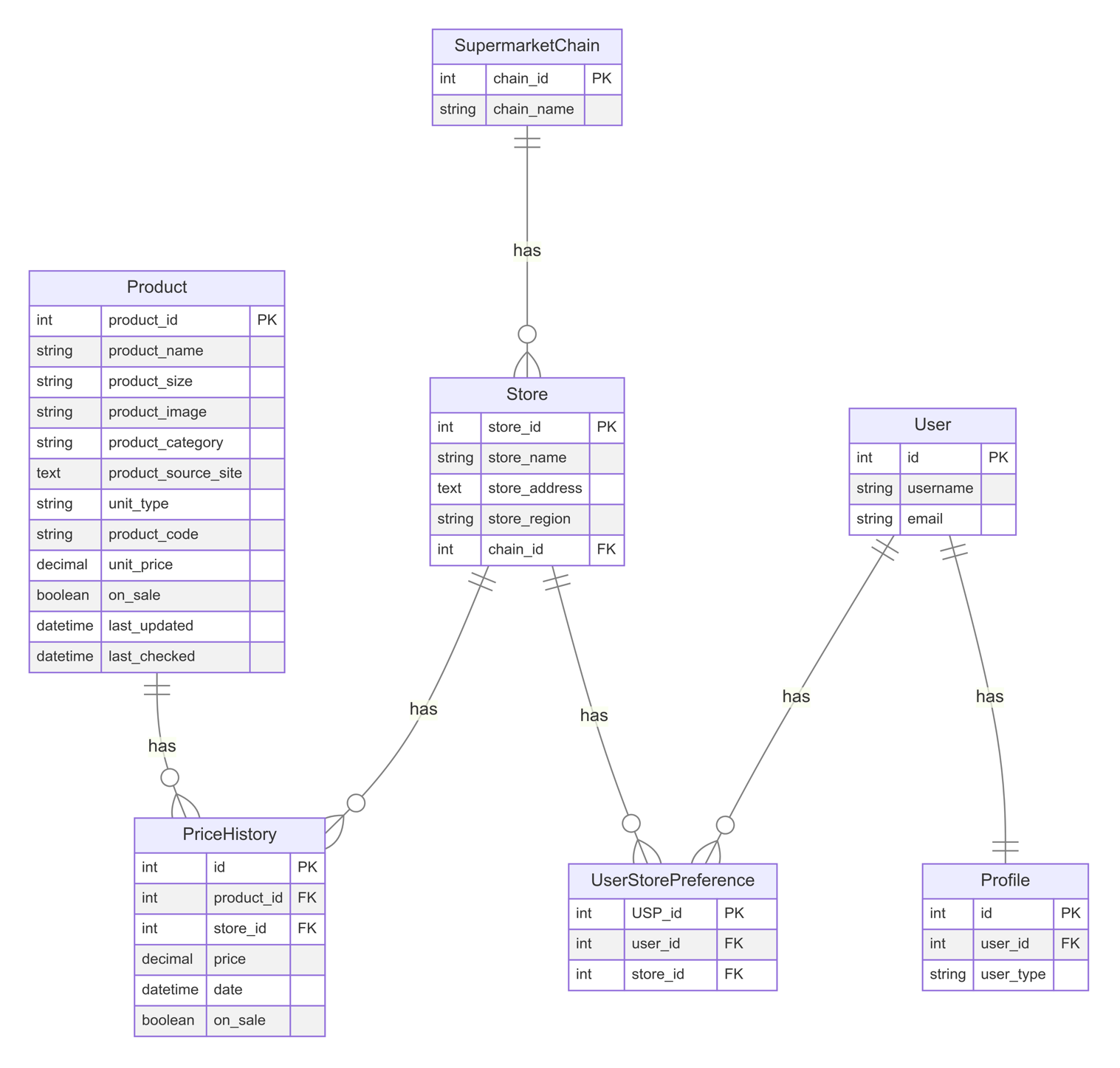
**Relational Database Design Version1.2**   
  
  
**Entities and Their Relationships**

SupermarketChain

- Primary Key: `chain\_id`

- Attributes:

- `chain\_name`: The name of the supermarket chain

Relationships:

- One-to-Many with Store: A supermarket chain can have multiple stores.

Store

- Primary Key: `store\_id`

- Attributes:

- `store\_name`: The name of the store

- `store\_address`: The address of the store

- `store\_region`: The region where the store is located

Relationships:

- Many-to-One with SupermarketChain: Each store belongs to one supermarket chain.

- Many-to-Many with Product: Stores have many products, and products can be in many stores (through PriceHistory).

- One-to-Many with PriceHistory: A store can have multiple price history records.

- Many-to-Many with User: Users can have preferences for multiple stores (through UserStorePreference).

Product

- Primary Key: `product\_id`

- Attributes:

- `product\_name`: The name of the product

- `product\_size`: The size of the product

- `product\_image`: An image of the product

- `product\_category`: The category of the product

- `product\_source\_site`: The source site of the product

- `unit\_type`: The type of unit for the product

- `product\_code`: A unique code for the product

- `unit\_price`: The price per unit of the product

- `on\_sale`: Indicates if the product is on sale

- `last\_updated`: The last update timestamp

- `last\_checked`: The last check timestamp

Relationships:

- Many-to-Many with Store: Products can be in many stores, and stores have many products (through PriceHistory).

- One-to-Many with PriceHistory: A product can have multiple price history records.

PriceHistory

- Primary Key: `id`

- Attributes:

- `price`: The price of the product

- `date`: The date of the price record

- `on\_sale`: Indicates if the product was on sale

Relationships:

- Many-to-One with Product: Each price history record belongs to one product.

- Many-to-One with Store: Each price history record is associated with one store.

User

- Primary Key: `id`

- Attributes:

- `username`: The username of the user

- `email`: The email address of the user

Relationships:

- One-to-One with Profile: Each user has one profile.

- Many-to-Many with Store: Users can have preferences for multiple stores (through UserStorePreference).

Profile

- Primary Key: `id`

- Attributes:

- `user\_type`: The type of user

Relationships:

- One-to-One with User: Each profile belongs to one user.

UserStorePreference

- Primary Key: `USP\_id`

Relationships:

- Many-to-One with User: Each preference belongs to one user.

- Many-to-One with Store: Each preference is associated with one store.

**Key Relationships Explained**

1. SupermarketChain and Store: This is a one-to-many relationship. A supermarket chain can have multiple stores, but each store belongs to only one supermarket chain. This allows for organising stores under their respective chains.

2. Store and Product: This is a many-to-many relationship implemented through the PriceHistory table. A store can have many products, and a product can be available in many stores. The PriceHistory table serves as a junction table, also storing historical price information.

3. Product and PriceHistory: This is a one-to-many relationship. A product can have multiple price history records, allowing for tracking price changes over time across different stores.

4. Store and PriceHistory: This is also a one-to-many relationship. A store can have multiple price history records for different products.

5. User and Profile: This is a one-to-one relationship. Each user has exactly one profile, and each profile belongs to exactly one user. This allows for storing additional user information separate from the main User model.

6. User, Store, and UserStorePreference: This is a many-to-many relationship between User and Store, implemented through the UserStorePreference table. It allows users to have preferences for multiple stores, and stores can be preferred by multiple users.