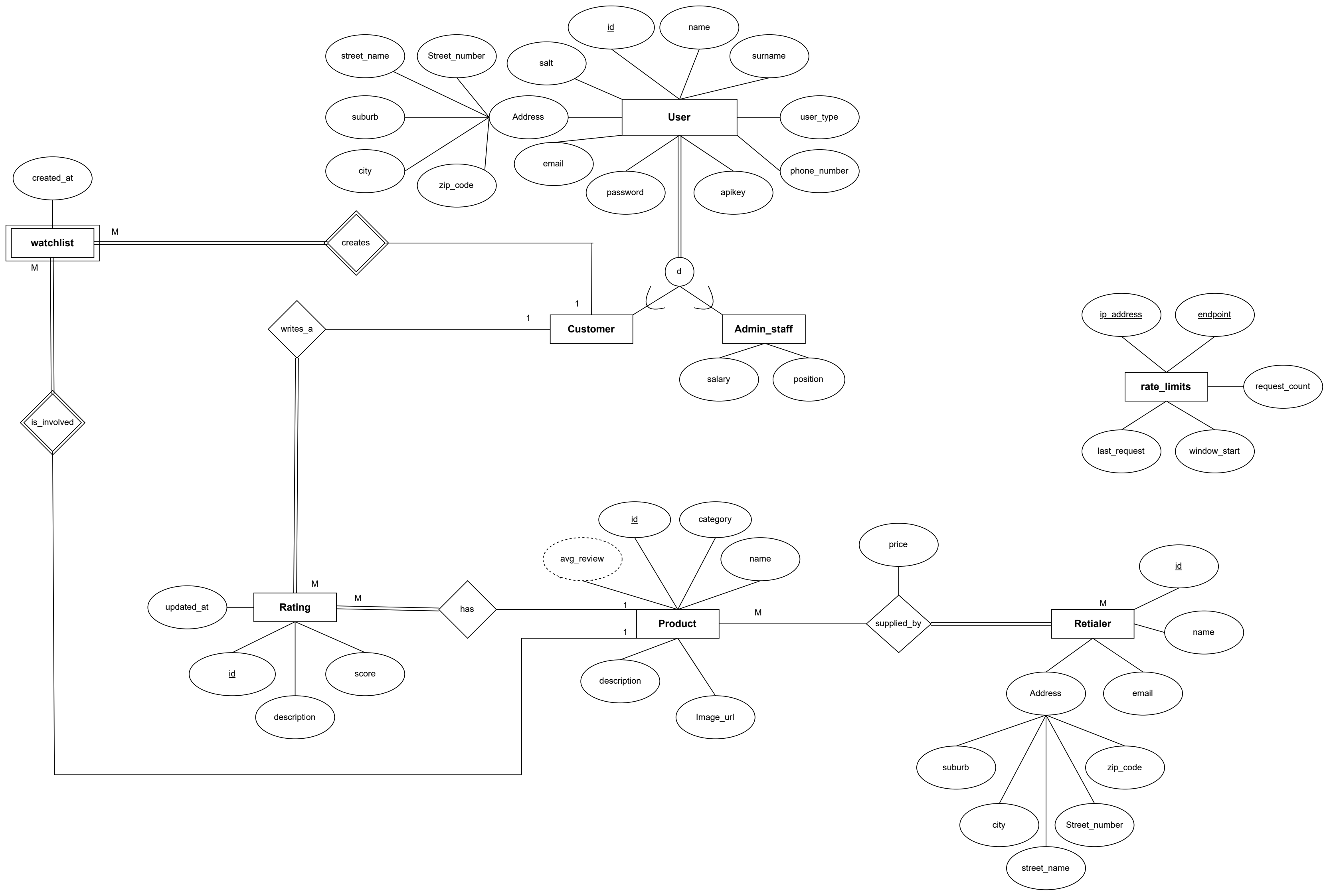

Comments:

- User is a strong entity with a disjoint specialization based on user_type. It has total participation: every User must be either a Customer or Admin_staff. Disjointness means that a user can be only one of the two, never both. A person in our system must have a role (either admin or customer), and cannot hold both roles simultaneously. This enforces clear access control logic (e.g., customers view and watchlist products, admins manage products, add retailers, etc.).

- avg_review is a derived attribute of Product. It is calculated from Rating entries (score) that are associated with this product.

- Watchlist is a weak entity: It cannot exist without both a Customer and a Product. It has no unique attribute to identify it alone. It is identified by a composite key (customer_id and product_id).

- Rating is a strong entity: It has its own primary key (id), and while it references both Customer and Product, it can exist independently.

- The rate_limits entity is a standalone system-level control table used to enforce rate-limiting mechanisms on the API. It is not part of the business (e.g., users, products, orders), but is essential to prevent API abuse. endpoint and ip_address form a composite primary key, uniquely identifying each tracked client-endpoint pair. This entity has no relationships to any other entities - its role is entirely operational/systemic.

Relationships:

- Customer creates Watchlist: One customer creates many watchlist entries. A customer might not create any watchlist entries. Every watchlist entry must be created by a customer. Customers can save products to their watchlist, but aren't required to

- Product is_involved Watchlist: One product can be in many watchlists. A product might not be in any watchlists. Every watchlist entry must reference a product. Products can be watched by multiple customers

- Customer writes_a Rating: One customer writes many ratings. A customer might not write any ratings. Every rating must be written by a customer. Customers can review products, but aren't required to

- Product has Rating: One product has many ratings. A product might not have any ratings yet. Every rating must be for a product. Products accumulate reviews over time

- Product supplied_by Retailer: Many products are supplied by many retailers. A product might not be supplied by any retailer. A retailer might not supply any products. Retailers carry inventory of products at specific prices. The price for this product as sol by this retailer is stored in a "price" attribute.