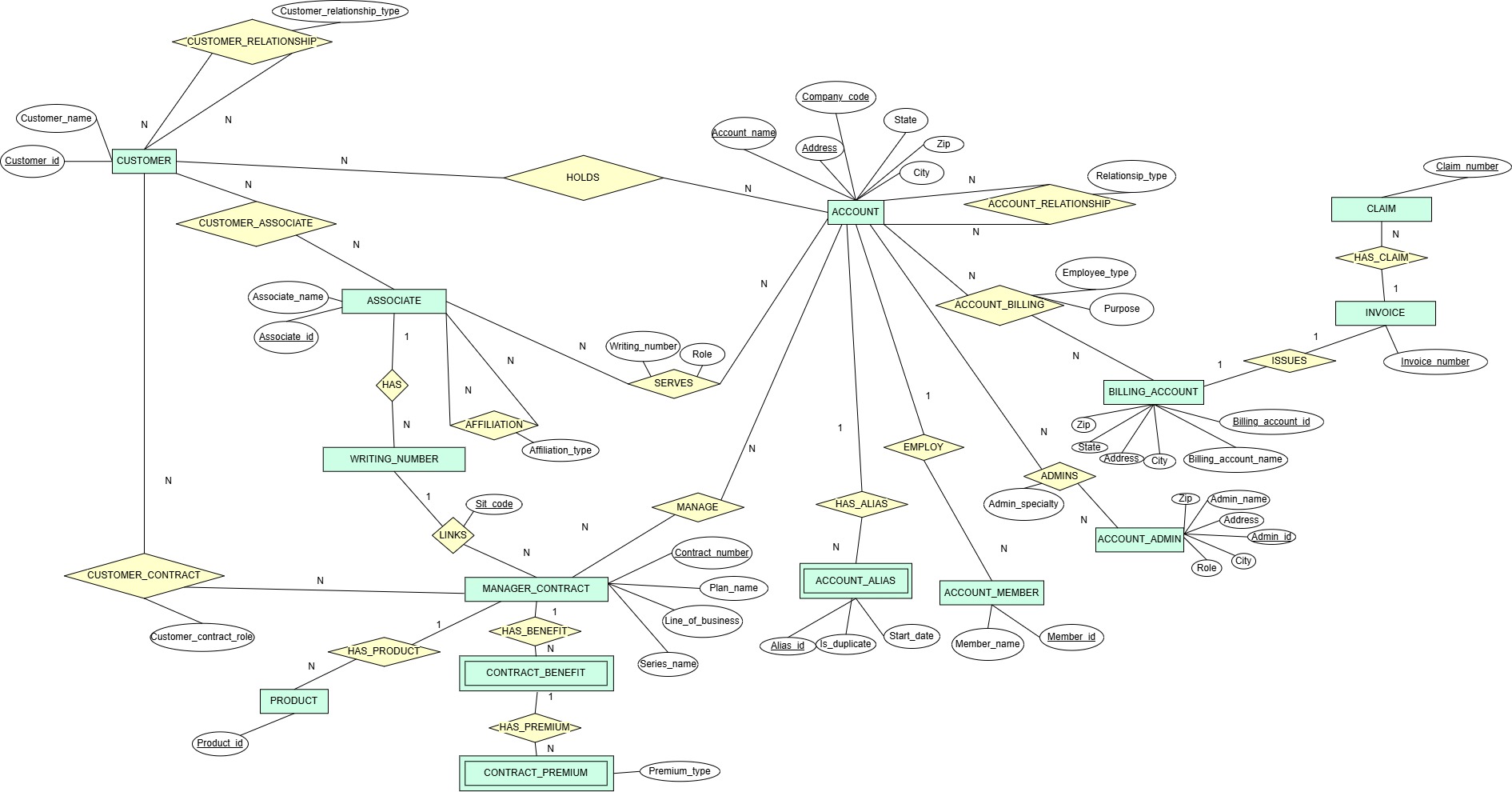
**1. ER Diagram:**



**2. Full Documentation**

**Entities and Primary Attributes**

The core entities in this ER diagram include Customer, Associate, Account, Writing\_Number, Manager\_Contract, Contract\_Benefit, Contract\_Premium, Account\_Admin, Billing\_Account, Account\_Alias, and Account\_Member.

Each entity has a unique primary key to ensure data integrity. For example, Customer\_id is the primary key for Customer, while Account uses a composite key (Account\_name, Address, Company\_code).

The Manager\_Contract entity, identified by Contract\_number, represents insurance contracts managed by associates and linked to specific benefits and premiums. Similarly, Account\_Admin and Billing\_Account are linked to Account for administrative and billing purposes, identified by Admin\_id and Billing\_account\_id, respectively.

The attributes within each entity capture essential information for business operations. For instance, Contract\_Premium includes Premium\_type (Commissions, Production Credit) to categorize premium payments. Account\_Alias includes Is\_duplicate, which flags duplicate entries, supporting account auditing and data integrity.

Additionally, Account\_Member represents individuals affiliated with an account, connected via the Account\_Member relationship, which tracks members or employees linked to specific accounts.

Invoice: Records information related to billing or customer transactions. Key attribute is Invoice\_number (primary key). Each invoice is associated with a Billing\_Account and may be linked to one or more claims, capturing transaction details and statuses.

Product: Represents various products or services offered by the enterprise. Key attribute is Product\_id (primary key). Products can be associated with Contracts, indicating specific products included within each contract.

Claim: Records claim information for invoices. Key attribute is Claim\_number (primary key). The Claim entity is linked to Invoice to track related claims.

**Key Relationships and Cardinalities**

Customer\_Relationship defines connections between customers, allowing for relationships such as family or trust-based associations, using Customer\_relationship\_type to clarify the nature of each relationship.

Customer\_Contract links Customer and Manager\_Contract with a Customer\_contract\_role attribute, indicating roles like Policy Owner or Beneficiary. The Holds relationship between Customer and Account reflects customers’ ownership of multiple accounts.

Associates are integral to this model, connected to Account and Manager\_Contract entities through several relationships. For instance, Serves connects Associate to Account with a Role attribute (e.g., Broker, Recruiter), defining their responsibilities.

Writing\_Number links to Associate via Affiliation, establishing business affiliations based on geographic or departmental distinctions.

HAS links Associate to Writing\_Number, supporting multiple writing numbers per associate.

**Contracts, Benefits, and Premiums**

The Manager\_Contract entity is central to the insurance offerings, connected to specific benefits and premiums.

The HAS\_BENEFIT relationship links Manager\_Contract to Contract\_Benefit, supporting one-to-many relationships to accommodate various benefit types within a contract.

Similarly, HAS\_PREMIUM connects Contract\_Benefit to Contract\_Premium, allowing each benefit to have multiple premium types.

**Administrative and Billing Functions**

To manage accounts, the model incorporates Account\_Admin and Billing\_Account entities.

The Employ relationship connects Account\_Admin with Account, capturing admin responsibilities with a Role attribute that specifies expertise, such as FSA administration.

Account\_Billing links Account to Billing\_Account entities, with attributes like Purpose and Employee\_type to differentiate between billing accounts used for various purposes (e.g., Life Insurance, Health Insurance) or employee categories.

**3. Case Verifications:**

**3.1 Cases Supporting Account Design Decisions**

* **Account (n..n) BillingAccount**

The Account\_Billing relationship links Account to Billing\_Account, with attributes like Purpose and Employee\_type. This structure supports multiple billing accounts for the same account based on different purposes (FSA, Life, A&H) or employee categories (salaried, hourly).

* **Account (n..n) Account**

The Account\_Relationship relationship, with Relationship\_type as an attribute, enables linking Account entities with various relationship types like GroupMaster and FlexMaster.

* **Account (n..n) AccountAdmin**

The Employ relationship between Account\_Admin and Account, with the Admin\_specialty attribute, allows each account to have multiple administrators with defined specialties.

* **Account (1..n) AccountMember**

The Account\_Member relationship enables accounts to have multiple members.

* **AccountAlias**

The HAS\_ALIAS relationship between Account and Account\_Alias allows the storage of alternate names with the Is\_duplicate attribute, supporting auditing and data cleansing.

* **Account (CompanyCode as part of the logical primary key)**

The inclusion of Company\_code as part of the Account primary key differentiates otherwise identical accounts by region (e.g., Columbus vs. NY groups at the same address).

**3.2 Cases Supporting Associate Design Decisions**

* **Account (n..n) ManagerContract**

The Manage relationship connects Associate and Manager\_Contract with Role attributes, allowing a single account to have multiple associates with different roles (Original Servicing, Assisting).

* **Associate (1..n) ManagerContract**

The Writing\_Number entity links to Associate through HAS and Affiliation relationships, with Affiliation\_type and Sit\_code attributes capturing formal and informal arrangements.

* **Associate Commissions vs. Production Credit**

The Contract\_Premium entity has a Premium\_type attribute that distinguishes between commission and production credit accounting, capturing both forms of associate compensation.

* **Associate (n..n) Associate**

The Affiliation relationship with Affiliation\_type attribute (Broker/ Recruiter) supports connections between associates, differentiating between Broker and Recruiter roles.

**3.3 Cases Supporting Contract Design Decisions**

* **Contract (1..n, 1..n) ContractPremium**

The HAS\_BENEFIT and HAS\_PREMIUM relationships link Manager\_Contract to Contract\_Benefit and then to Contract\_Premium, respectively. This structure supports multiple benefits per contract and multiple premium options for each benefit.

* 1. **Cases Supporting Customer Design Decisions**
* **Customer (n..n) Customer**

The Customer\_Relationship relationship links Customer entities with Customer\_relationship\_type, supporting family, beneficiary, and trust relationships.

* **Customer (n..n) Contract**

The Customer\_Contract relationship between Customer and Manager\_Contract includes the Customer\_contract\_role attribute, supporting various roles (Owner, Payer).

* **Customer (n..n) ContractBenefit**

The HAS\_BENEFIT relationship links Manager\_Contract to Contract\_Benefit, allowing specific benefits to be assigned to individual family members.

* **Customer (n..n) Account**

The Account\_Member relationship connects Customer to Account, supporting multiple affiliations and part-time arrangements.

* **Customer (n..n) Associate**

The Customer\_Associate relationship between Customer and Associate supports the transfer of commissions to family members after an associate's death.