

# Account Schema

## Account Entity

Column	Data Type	Description
account_id	UUID (PK)	Unique identifier for each account.
customer_id	UUID (FK)	Links to the Customer entity.
account_type	VARCHAR(50)	Enum for account categories (detailed below).
account_subtype	VARCHAR(50)	More granular classification within account types.
account_number	VARCHAR(20)	Unique account number (e.g., IBAN for international accounts).
currency_code	CHAR(3)	ISO 4217 currency code.
balance	DECIMAL(18,2)	Current available balance.
available_balance	DECIMAL(18,2)	Balance available for transactions (excludes holds, reserves).
open_date	DATE	Date the account was opened.
close_date	DATE	Date the account was closed (if applicable).
status	VARCHAR(20)	Enum: 'Active', 'Dormant', 'Closed', 'Frozen', 'Blocked'.
interest_rate	DECIMAL(5,2)	Interest rate applicable for savings, fixed deposit, and loans.
minimum_balance	DECIMAL(18,2)	Minimum required balance for certain accounts.
overdraft_limit	DECIMAL(18,2)	For current/checking accounts.
maturity_date	DATE	For fixed deposits and time-bound accounts.
account_holder_type	VARCHAR(10)	Enum: 'Individual', 'Joint', 'Corporate'.

linked_facility_id	UUID (FK)	Links to the CreditFacility table if applicable.
linked_branch_id	UUID (FK)	Links to the Branch entity.
is_international	BOOLEAN	Indicates if it's an international or domestic account.
account_category	VARCHAR(30)	High-level grouping (Retail, Corporate, Treasury, etc.).
created_by	UUID (FK)	Employee or system ID that created the account.
last_modified_by	UUID (FK)	Last person who updated the record.
created_at	TIMESTAMP	Record creation timestamp.
updated_at	TIMESTAMP	Last update timestamp.

## Account Types and Subtypes

Account Category	Account Type	Account Subtypes
<b>Retail</b>	Savings	Regular, Premium, Kids/Teens, Senior Citizens, High-Yield
	Checking	Personal, Joint, Student, Payroll
	Fixed Deposit	Short-Term, Long-Term, Recurring
	Loan	Home Loan, Auto Loan, Personal Loan, Student Loan
	Credit Card	Standard, Gold, Platinum, Business
	Investment	Brokerage, Mutual Fund, Retirement Savings
<b>Corporate</b>	Checking	Corporate Current, Business Transactional, Merchant Accounts
	Savings	Business Savings, Non-Profit Savings
	Fixed Deposit	Short-Term, Long-Term, Recurring

	Treasury/Cash Management	Liquidity Management, Trade Finance
	Corporate Loan	Working Capital Loan, Equipment Loan, Real Estate Loan
<b>Treasury/Wholesale</b>	Treasury Account	Cash Management, FX, Trade Finance
	Custodian Account	Securities Custody, Escrow Accounts
	Syndicated Loan	Participation, Agent Role
<b>Specialized</b>	Trust/Fiduciary Account	Estate Planning, Guardianship, Escrow
	Virtual/Digital Account	Virtual IBAN, E-wallet Account
	Charity/Non-profit	Registered Charity Accounts
<b>International</b>	Foreign Currency Account	Multi-Currency, NOSTRO, VOSTRO
	Offshore Account	For high-net-worth individuals and businesses

---

# Data pattern

## account\_id

The `account_id` in a retail banking system should be globally unique, secure, and easy to validate. Here are possible patterns for generating `account_id`:

---

### 1. Numeric Patterns (Sequential or Random)

Pattern	Example	Description
---------	---------	-------------

Sequential	10000000000 1	Incremented by 1 for each new account. Simple but predictable.
Timestamp-based	20250216000 1	YYYYMMDD + sequential suffix. Ensures chronological ordering.
Random Numeric	58321984567 2	Generated via a secure random number generator.
Partitioned Pattern	01-12345678 90	BranchCode-AccountSequence. Useful for regional segmentation.

Notes:

- Ensure uniqueness with proper indexing and collision checks.
- Sequential IDs can leak information about account volume.

2. Alphanumeric Patterns (Better for security and flexibility)

Pattern	Example	Description
UUID (Version 4)	550e8400-e29b-41d4-a716-446655440000	Universally unique, random.
Prefixed Identifier	ACC-202402-123456	Prefix for 'Account', date, and sequence.

Hash-Based (SHA256)	4a7d1ed414474e4033ac29cc3d3f5	Derived from customer & timestamp.
Base62 Encoded	X7H2ZL3K58	Compact, URL-safe, and case-sensitive.

Notes:

- UUID v4 is commonly used when uniqueness is critical.
- Hash-based IDs can embed contextual info (e.g., customer ID, timestamp).

3. Structured Banking-Specific Patterns (Encoding account type or branch)

Pattern	Example	Format
Branch-Based	1105-0001234567	BranchCode-AccountNumber
Product-Based	SAV-1234567890	AccountTypeCode-Sequence
Geo-Based	US-CA-1234567	CountryCode-StateCode-Sequence
Hybrid	ACC-SAV-1105-2025-000123	Combines product, branch, and year.

Notes:

- Facilitates operational insights but requires more parsing logic.
- Ensure no meaningful customer information is encoded.

---

#### 4. Specialized Patterns for International Banking

Pattern	Example	Usage
IBAN (International)	GB29NWBK60161331926819	International transactions (ISO 13616 standard).
SWIFT Code	NWBKGB2LXXX	Identifies banks for international transfers.
GL Account Code	ACC-GL-2024-001	For general ledger and internal accounts.

Notes:

- Use official formats for IBAN and SWIFT where applicable.
- Ensure validation logic adheres to international standards.

---

#### 5. Domain-Specific Customization (AI/ML-Driven)

- **Risk-Based Encoding:** Incorporates a risk profile or account segmentation into the ID.
- **AI-Generated Patterns:** Uses model-derived embeddings for uniqueness with contextual insights.

---

#### Recommended Best Practices

1. **Uniqueness:** Use UUID or hashing to avoid collisions.
2. **Security:** Avoid sequential IDs for external exposure.
3. **Flexibility:** Allow for extension to new account types.
4. **Performance:** Use indexed fields to optimize lookups.

# account\_number

The `account_number` in a retail banking system serves as a primary identifier for customer accounts. It should be unique, easily recognizable, and adhere to industry standards for certain account types (e.g., IBAN for international accounts).

## 1. Standard Numeric Patterns (Domestic)

Pattern	Example	Description
Simple Sequential	100000000001	Incremented for each new account; predictable but simple.
Branch-Based	1105-1234567890	BranchCode-AccountSequence. Useful for domestic operations.
Check-Digit	12345678-5	Last digit is a check digit for validation.
Date-Embedded	20250216-1234	YYYYMMDD + incremental suffix.
Random Numeric	583219845672	Secure random number generation.

Notes:

- Numeric-only patterns are typically used in domestic banking systems.
- Include a **check digit** using Luhn’s algorithm for data integrity.

## 2. International Patterns (Compliant with ISO standards)

Pattern	Example	Usage
IBAN (ISO 13616)	GB29NWBK60161331926819	International Bank Account Number.
SWIFT Code	NWBKGB2LXXX	SWIFT/BIC for cross-border transactions.
BBAN (Basic Bank Account Number)	60161331926819	Domestic part of the IBAN.

NOSTRO/VOSTRO	USD1234567890	For foreign currency accounts.
Offshore	SG-AC-9876543210	Offshore account with country prefix.

- Notes:
- IBAN structure varies by country but typically includes country code, check digits, bank identifier, and account number.
  - Use official IBAN validation rules when processing international transactions.
- 

3. Structured Patterns for Banking Products

Pattern	Example	Description
Product-Based	SAV-1234567890	Prefix with SAV for savings, CUR for current, etc.
Loan Accounts	LN-2025-0001234	Loan type (LN), year, and sequence number.
Investment Accounts	INV-4567890123	For brokerage and mutual fund accounts.
Credit Cards	4500 1234 5678 9012	Standard credit card pattern.
Corporate Accounts	CORP-0000012345	For corporate banking accounts.

- Notes:
- Product-based prefixes simplify account classification.
  - Ensure the pattern remains compatible with core banking systems.
- 

4. Security-Enhanced Patterns

Pattern	Example	Security Aspect
UUID (V4)	550e8400-e29b-41d4-a716-446655440000	Unpredictable and globally unique.



Hashed Identifier	a7d1ed414474e4033ac29cc3d3f5	Derived from customer ID, timestamp, or both.
Base62 Encoding	X7H2ZL3K58	Compact, non-sequential, URL-safe.
Tokenized Account Number	TKN-583219845672	Used for mobile wallets and payment gateways.

Notes:

- Prefer hashed or tokenized numbers for digital channels.
- UUIDs can be less user-friendly but are robust against enumeration attacks.

5. Institution-Specific Patterns

Pattern	Example	Usage
Bank Identifier Embedded	BANKX-1105-123456	Includes bank code for multi-entity systems.
Customer-Based	CUST1001-ACC56789	Combines customer ID and account sequence.
Geo-Based	US-NY-00012345	Encodes location for geographically segmented banking.
Joint Accounts	JOINT-ABCD123456	Used for accounts with multiple owners.

Notes:

- Useful for multi-bank systems or group-wide banking operations.

6. Hybrid & Customizable Patterns

Pattern	Example	Composition
Date-Product-Branch	2025-SAV-1105-001234	YYYY-PRODUCT-BRANCH-SEQUENCE.

Customer-Type-Prefix	IND-20240216-567890	CUSTOMER_TYPE-DATE-SEQUENCE.
Modulus-Based Account ID	ACC1234-MOD07	Suffix uses modulus 7 for integrity checks.
AI-Generated Segments	AI-TRX-58329X7L	Derived from historical transaction patterns.

**Notes:**

- Hybrid patterns balance structure, readability, and scalability.
- Useful for large banks offering diverse products.

---

**Best Practices for Account Numbers**

1. **Uniqueness:** Ensure account numbers are unique across the system.
2. **Predictability:** Avoid patterns that allow guessing of valid account numbers.
3. **Length:** Keep account numbers between 10–16 digits for domestic accounts, and follow IBAN standards for international ones.
4. **Validation:** Implement checksums (e.g., Luhn's algorithm) and pattern validation rules.
5. **Extensibility:** Choose patterns that can accommodate new products or geographic regions without changes to existing systems.

Would you like sample Python code to generate these patterns, or SQL DDL to enforce structure and uniqueness constraints?