Entity: Customer (Extended with Demographics)

Column Name	Data Type	Description
customer_id	VARCHAR(20)	Unique identifier for the customer (UUID or bank-generated).
entity_individual	VARCHAR(20)	can be text or reference
legal_company_name	VARCHAR(100)	
company_trade_name	VARCHAR(100)	
full_legal_name	VARCHAR(100)	
name_prefix	VARCHAR(10)	Mr, Mrs, Sri, Smt, Junior etc.
first_name	VARCHAR(100)	Customer’s first name.
middle_name	VARCHAR(100)	Customer’s middle name (nullable).
last_name	VARCHAR(100)	Customer’s last name.
alias	VARCHAR(100)	
date_of_birth	DATE	Customer’s date of birth.
date_of_incorporation	DATE	
national_id	VARCHAR(50)	National ID (e.g., SSN, Aadhar, Passport No.).

country_of_legal_formation	VARCHAR(100)	
country_of_birth	VARCHAR(100)	
country_of_principle_business	VARCHAR(100)	
country_of_primary_residence	VARCHAR(100)	
tax_id	VARCHAR(50)	Tax Identification Number (TIN).
age	INT	Customer's age (derived from date_of_birth).
gender	ENUM('Male', 'Female', 'Other', 'Prefer Not to Say')	Gender classification.
ethnicity	VARCHAR(100)	Customer's ethnic background (optional, for statistical insights).
nationality	VARCHAR(100)	Customer's country of citizenship.
citizenship1	VARCHAR(100)	
citizenship1	VARCHAR(100)	
native_language	VARCHAR(50)	Primary language spoken.
religion	VARCHAR(100)	Religious affiliation (optional).
marital_status	ENUM('Single', 'Married', 'Divorced', 'Widowed')	Marital status.
dependents_count	INT	Number of dependents (children, family members).

household_income	DECIMAL(15,2)	Total household income (if applicable).
education_level	ENUM('No Formal Education', 'High School', 'Associate Degree', 'Bachelor's Degree', 'Master's Degree', 'Doctorate')	Highest education level attained.
employment_status	ENUM('Employed', 'Self-employed', 'Unemployed', 'Retired', 'Student')	Employment status.
occupation	VARCHAR(100)	Customer's profession/job title.
industry	VARCHAR(100)	Industry the customer works in.
company_name	VARCHAR(255)	Employer's name (if employed).
years_with_employer	INT	Years of service with current employer.
job_role	VARCHAR(100)	Specific role at the company (e.g., Software Engineer, Accountant).
annual_income	DECIMAL(15,2)	Annual salary or business income.
primary_bank_relation	ENUM('Primary', 'Secondary')	Indicates if this is the customer's primary bank.
credit_score	DECIMAL(5,2)	Creditworthiness score (e.g., FICO, Experian).
home_ownership	ENUM('Owned', 'Rented', 'Mortgaged', 'Living with Family')	Housing situation.
residence_duration	INT	Number of years at current address.
vehicle_ownership	BOOLEAN	Indicates if the customer owns a vehicle.

financial_dependents	INT	Number of people financially dependent on the customer.
preferred_currency	VARCHAR(10)	Preferred currency for transactions.
preferred_branch	VARCHAR(100)	Closest or most frequently used bank branch.
preferred_contact_method	ENUM('Email', 'Phone', 'SMS', 'In-Person', 'Chatbot')	Preferred mode of communication.
social_media_presence	JSON	Social media handles (optional for AI profiling).
risk_score	DECIMAL(5,2)	AI-generated risk score (for underwriting & compliance).
created_at	TIMESTAMP	Date and time when the customer was added.
updated_at	TIMESTAMP	Last update timestamp.

Entity: Customer_Lifestyle

Captures personal interests, spending habits, and lifestyle choices.

Column Name	Data Type	Description
lifestyle_id	VARCHAR(20)	Unique lifestyle record ID.
customer_id	VARCHAR(20)	Reference to customer_id.
smoker	BOOLEAN	Indicates if the customer smokes.
alcohol_consumption	ENUM('Never', 'Occasionally', 'Regularly')	Frequency of alcohol consumption.

exercise_frequency	ENUM('None', '1-2 times/week', '3-5 times/week', 'Daily')	Exercise habits.
dietary_preferences	VARCHAR(255)	Vegan, vegetarian, gluten-free, etc.
travel_frequency	ENUM('Rarely', 'Occasionally', 'Frequently')	How often the customer travels.
luxury_spending	BOOLEAN	Indicates if the customer spends on luxury goods.
subscription_services	JSON	List of paid subscription services (Netflix, Amazon, gym, etc.).

Entity: Customer_Financial_Behavior

Tracks customer's financial habits and spending patterns.

Column Name	Data Type	Description
behavior_id	VARCHAR(20)	Unique behavior record ID.
customer_id	VARCHAR(20)	Reference to customer_id.
savings_habit	ENUM('Low', 'Moderate', 'High')	Savings behavior.
investment_type	ENUM('None', 'Stocks', 'Mutual Funds', 'Crypto', 'Bonds', 'Real Estate')	Type of investment.
loan_repayment_behavior	ENUM('Excellent', 'Good', 'Fair', 'Poor')	Payment track record.
spending_category	JSON	Categorized monthly spending (e.g., 30% Housing, 20% Food, 10% Travel).

preferred_payment_method	ENUM('Credit Card', 'Debit Card', 'UPI', 'Cash', 'Crypto')	Preferred mode of payment.
financial_goal	ENUM('Home Purchase', 'Retirement', 'Education', 'Vacation', 'Business')	Primary financial goal.

Entity: Customer_Digital_Engagement

Stores customer's interaction with digital banking channels.

Column Name	Data Type	Description
digital_id	VARCHAR(20)	Unique digital engagement ID.
customer_id	VARCHAR(20)	Reference to customer_id.
mobile_banking_usage	BOOLEAN	Indicates if customer uses mobile banking.
internet_banking_usage	BOOLEAN	Indicates if customer uses online banking.
preferred_digital_channel	ENUM('Web Portal', 'Mobile App', 'ATM', 'Branch')	Preferred mode of banking.
fraud_detection_enabled	BOOLEAN	Indicates if fraud alerts are activated.
biometric_authentication	BOOLEAN	Indicates if the customer uses fingerprint/face recognition.
recent_login	TIMESTAMP	Last login timestamp.

Entity: Customer_Address

Customers may have multiple addresses (home, work, etc.).

Column Name	Data Type	Description
address_id	VARCHAR(20)	Unique address ID.
customer_id	VARCHAR(20)	Reference to customer_id.
address_type	ENUM('Home', 'Work', 'Billing', 'Mailing')	Type of address.
street	VARCHAR(255)	Street address.
city	VARCHAR(100)	City.
state	VARCHAR(100)	State/Province.
postal_code	VARCHAR(20)	ZIP or postal code.
country	VARCHAR(100)	Country.
is_primary	BOOLEAN	Indicates if this is the primary address.

Entity: Customer_Contact

Holds customer contact details.

Column Name	Data Type	Description
contact_id	VARCHAR(20)	Unique contact ID.
customer_id	VARCHAR(20)	Reference to customer_id.
contact_type	ENUM('Mobile', 'Home Phone', 'Work Phone', 'Email')	Type of contact.
contact_value	VARCHAR(255)	Phone number or email.
is_primary	BOOLEAN	Indicates if this is the primary contact.

Data Pattern

Customer ID

---

1. UUID-Based (Universally Unique Identifier)

- Uses a **randomly generated UUID (v4)** or a **deterministic UUID (v5)**.
- **Pros**: Globally unique, works across distributed systems.
- **Cons**: Long and not human-friendly.

Pattern:
`xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx` (Hexadecimal, 36 characters including dashes)

Example Values:

- `550e8400-e29b-41d4-a716-446655440000`
- `a7b9c7d3-f0e2-4b7d-9b45-92e6d8c5b9a1`

---

2. Sequential Numeric ID

- Uses **incremental numbers** for simplicity.
- **Pros**: Easy to generate, short.
- **Cons**: Predictable (can be a security risk).

Pattern:
`CUST-0000001` (Fixed-length numeric sequence)

Example Values:

- `CUST-0000001`
- `CUST-1234567`
- `CUST-9999999`

---

3. Alphanumeric ID (Randomized)

- Uses **random alphanumeric** characters.
- **Pros**: More secure than sequential IDs.
- **Cons**: Harder to remember.

Pattern:
`CUST-XXXXXX` (6 uppercase letters/numbers)

Example Values:

- `CUST-A1B2C3`
- `CUST-X9Y8Z7`
- `CUST-MK2P4L`

---

4. Bank Branch + Customer Number

- Combines **branch code** and **sequential number**.
- **Pros**: Helps track customer location.
- **Cons**: Requires coordination between branches.

Pattern:
`BRANCHCODE-YYYY-NNNNNN`
(Branch Code + Year + Sequential ID)

Example Values:

- `NYC-2025-000123`
- `LDN-2024-987654`
- `DEL-2023-001234`

---

5. Date-Based ID

- Uses **date of registration** + **sequence number**.
- **Pros**: Easy to track sign-up dates.
- **Cons**: Limited uniqueness if only using date.

Pattern:
`YYYYMMDD-XXXXX` (Date + 5-digit sequence)

Example Values:

- `20250210-12345`
- `20240101-67890`
- `20231231-54321`

---

6. Phone Number-Based ID

- Uses the **last N digits** of a customer's phone number.
- **Pros**: Easier for customers to remember.
- **Cons**: Can lead to duplicates.

Pattern:
`CUST-Last6DigitsOfPhone`

Example Values:

- `CUST-987654` (Phone: +1-555-123-987654)
- `CUST-456789` (Phone: +44-777-456789)

---

7. Country Code + Sequential ID

- Uses **ISO country code** + **incremental number**.
- **Pros**: Helps with international banking.
- **Cons**: Requires centralized tracking.

Pattern:
`CC-NNNNNNNN` (Country Code + Sequential)

Example Values:

- `US-00001234`
- `IN-98765432`
- `UK-12345678`

---

8. Initials + Birth Year + Random Digits

- Uses **customer’s initials**, **birth year**, and **random digits**.

- **Pros**: Personalized but still unique.
- **Cons**: Can still lead to duplicates.

Pattern:
`XXYY-XXXXX` (Initials + Birth Year + Random)

Example Values:

- `JD85-12345` (John Doe, born in 1985)
- `AM92-67890` (Alice Morgan, born in 1992)

---

9. Encrypted ID (Hashed)

- Uses **hashing (e.g., SHA256, MD5, or Base64 encoding)**.
- **Pros**: Highly secure.
- **Cons**: Hard to manage manually.

Pattern:
`Base64(Hash(Customer Data))`

Example Values:

- `5d41402abc4b2a76b9719d911017c592` (MD5 hash)
- `dGhpcyBpcyBhIHRlc3QgY3VzdG9tZXI=` (Base64 encoded)

---

10. Smart Card / RFID-Based ID

- Uses a **smart card chip ID** or **RFID tag number**.
- **Pros**: Works well for physical banking cards.
- **Cons**: Needs integration with hardware.

Pattern:
`RFID-XXXXXXXXXXXXXXXX` (Hexadecimal format)

Example Values:

- `RFID-1A2B3C4D5E6F7G8H`
- `RFID-9F8E7D6C5B4A3F2E`

---

11. QR Code / Barcode-Based ID

- Uses a **QR Code** or **barcode** that encodes a unique string.
- **Pros**: Ideal for mobile banking.
- **Cons**: Requires scanning tools.

Pattern:
`QR-ENCODED-CUSTOMER-ID`

Example Values (Decoded):

- `QR-CUST1234567890`
- `QR-550e8400e29b41d4a716446655440000`

Date of Birth

The `date_of_birth` (`DOB`) field in a retail banking system can follow **various formats** depending on the **system, locale, and storage requirements**. Below are **all possible patterns** for storing and displaying `date_of_birth`, along with **examples**.

---

1. Standard SQL Date Format (ISO 8601)

- **Format:** `YYYY-MM-DD`
- **Example:** `1985-07-23`
- **Usage:** Preferred for **databases** (MySQL, PostgreSQL, Oracle, SQL Server).
- **Pros:** Easy to sort, globally accepted.
- **Cons:** Not human-friendly in some regions.

---

2. Full Date with Time (Timestamp)

- **Format:** `YYYY-MM-DD HH:MI:SS`
- **Example:** `1985-07-23 00:00:00`

- **Usage:** Used when the **exact birth time** is needed (e.g., astrology-based financial insights).
- **Pros:** Includes time if required.
- **Cons:** Unnecessary time component for most banking applications.

---

3. Slash-Separated (Common in US, UK)

- **Format:** `MM/DD/YYYY`
- **Example:** `07/23/1985`
- **Usage: User-friendly format in the US, UK.**
- **Pros:** Easy to read for Americans.
- **Cons: Ambiguous** in global applications (07/08/1985 → July 8 or August 7?).

---

4. European Format (Day First)

- **Format:** `DD/MM/YYYY`
- **Example:** `23/07/1985`
- **Usage:** Common in **Europe, Latin America, Asia**.
- **Pros:** Matches spoken language order.
- **Cons:** Confusing for American users.

---

5. Hyphen-Separated (Alternative)

- **Format:** `DD-MM-YYYY`
- **Example:** `23-07-1985`
- **Usage:** Common in **Europe, India, Middle East**.
- **Pros:** Readable, avoids `/` confusion.
- **Cons:** May not be the default for databases.

---

6. Short Year Format

- **Format:** `DD-MM-YY`
- **Example:** `23-07-85`

- **Usage:** Used in **old banking systems** with legacy constraints.
- **Pros:** Saves storage space.
- **Cons: Ambiguous century** (1985 or 2085?).

---

7. Month Abbreviation (Text-Based)

- **Format:** `DD-MMM-YYYY`
- **Example:** `23-Jul-1985`
- **Usage: Human-friendly format** used in reports.
- **Pros:** Eliminates month confusion.
- **Cons:** Can be inconsistent in different languages (`Jul` vs. `Juillet` in French).

---

8. Full Month Name (Formal)

- **Format:** `DD Month YYYY`
- **Example:** `23 July 1985`
- **Usage:** Used in **formal documents** and statements.
- **Pros: Very clear** to humans.
- **Cons:** Takes **more space**, harder for computers to parse.

---

9. Year-Month Only (For Approximate Data)

- **Format:** `YYYY-MM`
- **Example:** `1985-07`
- **Usage:** When **exact date is unknown** (e.g., customer only remembers birth year & month).
- **Pros:** Useful for **approximate birthdates**.
- **Cons:** Lacks full precision.

---

10. Year Only (For Privacy Reasons)

- **Format:** `YYYY`
- **Example:** `1985`

- **Usage:** Used when **only the birth year is needed** (e.g., age-based services).
- **Pros: Good for anonymization**.
- **Cons:** Cannot determine exact age.

---

11. Unix Timestamp (Epoch Time)

- **Format: Number of seconds since January 1, 1970**
- **Example:** `492134400` (`1985-07-23 00:00:00 UTC`)
- **Usage:** Used in **computing & APIs**.
- **Pros:** Ideal for **calculations & time zones**.
- **Cons:** Not human-readable.

---

12. Julian Date (Astronomical, Military Use)

- **Format:** `YYYYDDD`
- **Example:** `1985204` (204th day of 1985 = July 23, 1985)
- **Usage:** Used in **historical records, astronomy, military banking**.
- **Pros:** Eliminates month complexity.
- **Cons:** Hard for humans to interpret.

---

13. Bank-Specific Alphanumeric Code

- **Format:** `YYMMDDXXXX`
- **Example:** `850723A12B`
- **Usage:** Internal **banking identifiers**.
- **Pros:** Unique per customer.
- **Cons:** Hard to interpret.

National ID

---

1. Pure Numeric (Fixed Length)

- **Format:** `NNNNNNNNN` (Fixed-length numbers)
- **Example:** `1234567890`
- **Usage: USA (SSN), India (Aadhaar), China (Resident ID)**
- **Pros:** Simple, compact.
- **Cons:** Lacks structure for validation.

Examples by Country

Country	Format	Example
USA (SSN)	`NNN-NN-NNNN`	`123-45-6789`
India (Aadhaar)	`NNNN-NNNN-NNNN`	`1234-5678-9012`
China (Resident ID)	`NNNNNNYYYYMMDDNNNN`	`110105199003072118`

---

2. Alphanumeric (Fixed Length)

- **Format:** `AANNNNNNN`
- **Example:** `AB1234567`
- **Usage: UK (NIN), Australia (TFN), Canada (SIN)**
- **Pros:** More unique combinations.
- **Cons:** Harder to remember.

Examples by Country

Country	Format	Example
UK (NIN)	`AA-NNNNNN-A`	`QQ-123456-C`

Canada (SIN)	`NNN-NNN-NNN`	`123-456-789`
Australia (TFN)	`NNN-NNN-NNN`	`123-456-789`

---

3. Country Code + ID Number

- **Format:** `CC-NNNNNNNNN`
- **Example:** `IN-1234567890`
- **Usage: International banking systems**
- **Pros:** Helps identify the country.
- **Cons:** Requires global coordination.

Example

Country	Format	Example
India Aadhaar (with country)	`IN-1234-5678-9012`	`IN-1234-5678-9012`
USA SSN (with country)	`US-123-45-6789`	`US-123-45-6789`

---

4. Segmented Numeric Pattern

- **Format:** `NNN-NNN-NNN`
- **Example:** `123-456-789`
- **Usage: USA (SSN), Canada (SIN), Australia (TFN)**
- **Pros:** Easier to read.
- **Cons:** Needs strict validation.

Examples by Country

Country	Format	Example
USA (SSN)	`NNN-NN-NNNN`	`123-45-6789`
Canada (SIN)	`NNN-NNN-NNN`	`987-654-321`
Australia (TFN)	`NNN-NNN-NNN`	`123-456-789`

---

5. Alphanumeric with Special Characters

- **Format:** `AA-NNNNNNN-A`
- **Example:** `AB-123456-C`
- **Usage: UK (National Insurance), Germany (Tax ID)**
- **Pros:** More unique.
- **Cons:** Hard to remember.

Examples by Country

Country	Format	Example
UK (NIN)	`AA-NNNNNN-A`	`QQ-123456-C`
Germany (Steuer-ID)	`NNNNNNNNNNN`	`12345678901`

---

6. Date-Based ID

- **Format:** `YYYYMMDD-NNNN`

- **Example:** `19900307-2118`
- **Usage: China, Sweden, South Africa**
- **Pros:** Easy age verification.
- **Cons:** Privacy concerns.

Examples by Country

Country	Format	Example
China (Resident ID)	`NNNNNNYYYYMMDDNNNN`	`110105199003072118`
Sweden (Personal Number)	`YYYYMMDD-NNNN`	`19850723-1234`
South Africa (ID Number)	`YYMMDDSSSSCAZ`	`8507231234083`

---

7. Hexadecimal / Encrypted Format

- **Format:** `XXXXXXXX`
- **Example:** `A1B2C3D4`
- **Usage: Smart cards, blockchain-based identity**
- **Pros:** Highly secure.
- **Cons:** Hard to verify manually.

---

8. QR Code / Smart Card ID

- **Format:** `QR-CODE-ENCODED-ID`
- **Example:** QR Code that represents `1234567890`
- **Usage: Digital banking and mobile banking**
- **Pros:** Contactless, scannable.
- **Cons:** Requires hardware support.

---

9. RFID / Smart Chip ID

- **Format:** `RFID-XXXXXXXXXXXXXXXX`
- **Example:** `RFID-1A2B3C4D5E6F7G8H`
- **Usage: Digital IDs embedded in cards**
- **Pros:** Works well for biometric authentication.
- **Cons:** Hardware-dependent.

Tax ID

---

1. Pure Numeric (Fixed Length)

- **Format:** `NNNNNNNNN`
- **Example:** `123456789`
- **Usage: USA (TIN, EIN, SSN), Canada (SIN), India (PAN)**
- **Pros:** Simple, compact.
- **Cons:** Lacks structure for validation.

Examples by Country

Country	Format	Example
USA (TIN/EIN/SSN)	`NNN-NN-NNNN`	`123-45-6789`
Canada (SIN)	`NNN-NNN-NNN`	`123-456-789`
Germany (Steuer-ID)	`NNNNNNNNNNN`	`12345678901`

---

2. Alphanumeric (Fixed Length)

- **Format:** `AANNNNNNN`
- **Example:** `AB1234567`
- **Usage: India (PAN), UK (UTR), Australia (ABN)**
- **Pros:** More unique combinations.
- **Cons:** Harder to remember.

Examples by Country

Country	Format	Example
India (PAN)	`AAAAA9999A`	`ABCDE1234F`
UK (UTR)	`NNNNNNNNNN`	`1234567890`
Australia (ABN)	`NN NNN NNN NNN`	`12 345 678 901`

---

3. Country Code + Tax Number

- **Format:** `CC-NNNNNNNNN`
- **Example:** `US-123456789`
- **Usage: International taxation**
- **Pros:** Helps identify the country.
- **Cons:** Requires global coordination.

Example

Country	Format	Example
India PAN (with country)	`IN-ABCDE1234F`	`IN-ABCDE1234F`

USA TIN (with country)	`US-123-45-6789`	`US-123-45-6789`

---

4. Segmented Numeric Pattern

- **Format:** `NNN-NN-NNNN`
- **Example:** `123-45-6789`
- **Usage: USA (TIN, SSN), Canada (SIN)**
- **Pros:** Easier to read.
- **Cons:** Needs strict validation.

Examples by Country

Country	Format	Example
USA (TIN/SSN)	`NNN-NN-NNNN`	`123-45-6789`
Canada (SIN)	`NNN-NNN-NNN`	`987-654-321`

---

5. Alphanumeric with Special Characters

- **Format:** `AA-NNNNNNN-A`
- **Example:** `AB-123456-C`
- **Usage: UK (UTR), Germany (Tax ID)**
- **Pros:** More unique.
- **Cons:** Hard to remember.

Examples by Country

Country	Format	Example

UK (UTR)	`NNNNNNNNNN`	`1234567890`
Germany (Steuer-ID)	`NNNNNNNNNNN`	`12345678901`

---

6. Date-Based ID

- **Format:** `YYYYMMDD-NNNN`
- **Example:** `19900307-2118`
- **Usage: China, Sweden, South Africa**
- **Pros:** Easy age verification.
- **Cons:** Privacy concerns.

Examples by Country

Country	Format	Example
China (Resident Tax ID)	`NNNNNNYYYYMMDDNNNN`	`110105199003072118`
Sweden (Tax ID)	`YYYYMMDD-NNNN`	`19850723-1234`

---

7. Hexadecimal / Encrypted Format

- **Format:** `XXXXXXXX`
- **Example:** `A1B2C3D4`
- **Usage: Smart cards, blockchain-based taxation**
- **Pros:** Highly secure.
- **Cons:** Hard to verify manually.

---

8. QR Code / Digital Tax ID

- **Format:** `QR-CODE-ENCODED-ID`
- **Example:** QR Code that represents `123456789`
- **Usage: Digital taxation systems**
- **Pros:** Contactless, scannable.
- **Cons:** Requires hardware support.
