Database Schema

Bilko Database Schema

> **Status:** IMPLEMENTED (Prisma schema exists) **Location:** `/Users/makinja/ALAI/products/Bilko/packages/database/prisma/schema.prisma`
> **Database:** PostgreSQL 14+ **ORM:** Prisma 5.x **Last updated:** 2026-02-20

Purpose

This document describes the complete database schema for Bilko. The schema is IMPLEMENTED in Prisma and ready for migration. This doc explains the relationships, constraints, and design decisions.

Entity Relationship Overview

```
Organization (1) ──┬── (N) User
                   ├── (N) Account
                   ├── (N) Contact
                   ├── (N) Invoice
                   ├── (N) Expense
                   ├── (N) Transaction
                   └── (N) BankAccount

Contact (1) ──────┬── (N) Invoice
                  └── (N) Expense

Invoice (1) ──── (N) InvoiceItem

Account (1) ──┬── (N) InvoiceItem
```

```
                ├── (N) Expense
                ├── (N) BankAccount
                ├── (N) Transaction (debit)
                ├── (N) Transaction (credit)
                └── (N) Account (parent-child hierarchy)

BankAccount (1) ── (N) BankTransaction

Currency (1) ──┬── (N) ExchangeRate (base)
               └── (N) ExchangeRate (target)

User (1) ──┬── (N) Invoice (creator)
           ├── (N) Expense (creator)
           ├── (N) Expense (approver)
           ├── (N) Transaction (creator)
           └── (N) LoggedAction
```

Full ER Diagram

```
erDiagram
    Organization {
        UUID id PK
        VARCHAR name
        VARCHAR registrationNumber
        VARCHAR vatNumber
        CHAR baseCurrency
        CHAR country
        CHAR language
        DATE fiscalYearStart
        TIMESTAMP createdAt
        TIMESTAMP updatedAt
    }

    User {
        UUID id PK
        UUID organizationId FK
        VARCHAR email
        VARCHAR passwordHash
        VARCHAR fullName
```

```
    ENUM role
    BOOLEAN twoFactorEnabled
    VARCHAR twoFactorSecret
    TIMESTAMP lastLoginAt
    TIMESTAMP createdAt
    TIMESTAMP updatedAt
}

AccountType {
    INT id PK
    VARCHAR name
    ENUM normalBalance
    TIMESTAMP createdAt
}

Account {
    UUID id PK
    UUID organizationId FK
    VARCHAR code
    VARCHAR name
    INT accountTypeId FK
    CHAR currencyCode
    UUID parentAccountId FK
    BOOLEAN isActive
    TIMESTAMP createdAt
    TIMESTAMP updatedAt
}

Contact {
    UUID id PK
    UUID organizationId FK
    ENUM type
    VARCHAR name
    VARCHAR email
    VARCHAR phone
    VARCHAR vatNumber
    VARCHAR addressLine1
    VARCHAR city
    CHAR country
    CHAR currencyCode
```

```
    INT paymentTerms
    BOOLEAN isActive
    TIMESTAMP createdAt
    TIMESTAMP updatedAt
}

Invoice {
    UUID id PK
    UUID organizationId FK
    UUID customerId FK
    VARCHAR invoiceNumber
    DATE invoiceDate
    DATE dueDate
    CHAR currencyCode
    DECIMAL exchangeRate
    DECIMAL subtotal
    DECIMAL taxAmount
    DECIMAL discountAmount
    DECIMAL totalAmount
    DECIMAL baseAmount
    ENUM status
    TIMESTAMP sentAt
    TIMESTAMP paidAt
    VARCHAR pdfUrl
    UUID createdBy FK
    TIMESTAMP createdAt
    TIMESTAMP updatedAt
}

InvoiceItem {
    UUID id PK
    UUID invoiceId FK
    INT lineNumber
    VARCHAR description
    DECIMAL quantity
    DECIMAL unitPrice
    DECIMAL taxRate
    DECIMAL lineTotal
    UUID accountId FK
    TIMESTAMP createdAt
```

```
}

Expense {
    UUID id PK
    UUID organizationId FK
    UUID vendorId FK
    VARCHAR expenseNumber
    DATE expenseDate
    CHAR currencyCode
    DECIMAL exchangeRate
    DECIMAL amount
    DECIMAL baseAmount
    DECIMAL taxAmount
    VARCHAR category
    VARCHAR paymentMethod
    UUID accountId FK
    ENUM status
    UUID approvedBy FK
    TIMESTAMP approvedAt
    UUID createdBy FK
    TIMESTAMP createdAt
    TIMESTAMP updatedAt
}

Transaction {
    UUID id PK
    UUID organizationId FK
    DATE transactionDate
    VARCHAR description
    UUID debitAccountId FK
    UUID creditAccountId FK
    DECIMAL amount
    CHAR currencyCode
    DECIMAL exchangeRate
    DECIMAL baseAmount
    VARCHAR referenceType
    UUID referenceId
    BOOLEAN locked
    BOOLEAN reconciled
    UUID createdBy FK
```

```
    TIMESTAMP createdAt
}

BankAccount {
    UUID id PK
    UUID organizationId FK
    UUID accountId FK
    VARCHAR bankName
    VARCHAR accountNumber
    VARCHAR iban
    CHAR currencyCode
    DECIMAL currentBalance
    BOOLEAN isActive
    TIMESTAMP createdAt
    TIMESTAMP updatedAt
}

BankTransaction {
    UUID id PK
    UUID bankAccountId FK
    DATE transactionDate
    DECIMAL amount
    VARCHAR description
    VARCHAR reference
    BOOLEAN reconciled
    UUID matchedTransactionId
    TIMESTAMP createdAt
}

Currency {
    CHAR code PK
    VARCHAR name
    VARCHAR symbol
    SMALLINT decimalPlaces
    BOOLEAN isActive
    TIMESTAMP createdAt
}

ExchangeRate {
    UUID id PK
```

```
    CHAR baseCurrency FK
    CHAR targetCurrency FK
    DECIMAL rate
    DATE effectiveDate
    VARCHAR source
    TIMESTAMP lastUpdated
}

LoggedAction {
    BIGINT eventId PK
    TEXT schemaName
    TEXT tableName
    UUID userId FK
    TIMESTAMP actionTimestamp
    ENUM action
    JSONB rowData
    JSONB changedFields
    INET clientIp
}

SchemaVersion {
    VARCHAR version PK
    TIMESTAMP appliedAt
    TEXT description
}

Organization ||--o{ User : "has"
Organization ||--o{ Account : "owns"
Organization ||--o{ Contact : "manages"
Organization ||--o{ Invoice : "issues"
Organization ||--o{ Expense : "tracks"
Organization ||--o{ Transaction : "records"
Organization ||--o{ BankAccount : "holds"

User ||--o{ Invoice : "createdBy"
User ||--o{ Expense : "createdBy"
User ||--o{ Expense : "approvedBy"
User ||--o{ Transaction : "createdBy"
User ||--o{ LoggedAction : "performed"
```

```
    AccountType ||--o{ Account : "categorizes"
    Account ||--o{ Account : "parentOf"
    Account ||--o{ InvoiceItem : "revenueAccount"
    Account ||--o{ Expense : "expenseAccount"
    Account ||--o| BankAccount : "glAccount"
    Account ||--o{ Transaction : "debitAccount"
    Account ||--o{ Transaction : "creditAccount"

    Contact ||--o{ Invoice : "customer"
    Contact ||--o{ Expense : "vendor"

    Invoice ||--o{ InvoiceItem : "contains"

    BankAccount ||--o{ BankTransaction : "has"

    Currency ||--o{ ExchangeRate : "base"
    Currency ||--o{ ExchangeRate : "target"
```

Multi-Tenant Scoping

```
graph LR
    ORG[Organization\nMulti-tenant Root]

    ORG --> U[Users\nowner/admin/accountant/viewer]
    ORG --> COA[Chart of Accounts\nHierarchical GL]
    ORG --> C[Contacts\nCustomers & Vendors]
    ORG --> INV[Invoices\nOutgoing]
    ORG --> EXP[Expenses\nIncoming]
    ORG --> TXN[Transactions\nDouble-Entry Ledger]
    ORG --> BANK[BankAccounts\nReconciliation]

    INV --> ITEM[InvoiceItems\nLine Items]
    BANK --> BTXN[BankTransactions\nStatement Import]
    TXN --> LOG[LoggedAction\nAudit Trail]

    style ORG fill:#00E5A0,color:#000
    style TXN fill:#ffd700,color:#000
```

Core Tables

1. Organization

Purpose: Multi-tenant root. Every business is one organization.

Column	Type	Constraints	Description
id	UUID	PK, default uuid_generate_v4()	Primary key
name	VARCHAR(255)	NOT NULL	Business name
registrationNumber	VARCHAR(50)	NULL	Company tax ID
vatNumber	VARCHAR(50)	NULL	VAT registration number
baseCurrency	CHAR(3)	NOT NULL, default 'EUR'	ISO 4217 currency code
country	CHAR(2)	NOT NULL	ISO 3166-1 alpha-2 country code
language	CHAR(2)	NOT NULL, default 'sr'	ISO 639-1 language code
fiscalYearStart	DATE	NOT NULL, default '2026-01-01'	Fiscal year start date
createdAt	TIMESTAMP	NOT NULL, default now()	Record creation timestamp
updatedAt	TIMESTAMP	NOT NULL, default now()	Last update timestamp

Indexes:

- Primary key: `id`

Business rules:

- baseCurrency determines default currency for all transactions
- country determines tax rules (Serbia 20%, BiH 17%, Croatia 25%)
- fiscalYearStart used for annual reports

---

2. User

Purpose: Users within an organization. Role-based access control.

Column	Type	Constraints	Description
id	UUID	PK	Primary key

Column	Type	Constraints	Description
organizationId	UUID	FK → Organization, NOT NULL, CASCADE	Organization membership
email	VARCHAR(255)	UNIQUE, NOT NULL	Login email
passwordHash	VARCHAR(255)	NOT NULL	bcrypt hash (12 rounds)
fullName	VARCHAR(255)	NOT NULL	Display name
role	ENUM	NOT NULL	owner, admin, accountant, viewer
twoFactorEnabled	BOOLEAN	NOT NULL, default false	2FA status
twoFactorSecret	VARCHAR(255)	NULL	TOTP secret
lastLoginAt	TIMESTAMP	NULL	Last login timestamp
createdAt	TIMESTAMP	NOT NULL	Account creation
updatedAt	TIMESTAMP	NOT NULL	Last update

Indexes:

- Primary key: `id`
- Unique: `email`
- Foreign key: `organizationId` → Organization(id)
- Index: `idx_users_organization` on organizationId
- Index: `idx_users_email` on email

Enums:

```
enum UserRole {
  owner       // Full access, can delete org
  admin       // Can manage users and settings
  accountant  // Can create invoices/expenses
  viewer      // Read-only access
}
```

Business rules:

- One owner per organization (enforced in API)
- Cannot delete owner
- Password must be bcrypt hashed, NEVER plain text

Chart of Accounts

Account Hierarchy & Normal Balances

```
graph TD
    COA[Chart of Accounts]

    COA --> A[1xxx Assets\nnormalBalance: debit]
    COA --> L[2xxx Liabilities\nnormalBalance: credit]
    COA --> E[3xxx Equity\nnormalBalance: credit]
    COA --> R[4xxx Revenue\nnormalBalance: credit]
    COA --> EX[5xxx Expenses\nnormalBalance: debit]

    A --> CA[1100 Current Assets]
    A --> FA[1500 Fixed Assets]
    CA --> Cash[1110 Cash]
    CA --> Bank[1120 Bank Accounts]
    CA --> AR[1200 Accounts Receivable]
    Bank --> B1[1121 Intesa RSD]
    Bank --> B2[1122 Raiffeisen EUR]

    L --> CL[2100 Current Liabilities]
    L --> LL[2500 Long-term]
    CL --> AP[2110 Accounts Payable]
    CL --> VAT[2120 VAT Payable]

    E --> SC[3100 Share Capital]
    E --> RE[3900 Retained Earnings]

    R --> SR[4100 Service Revenue]
    R --> PR[4200 Product Sales]

    EX --> OE[5100 Operating Expenses]
    OE --> SAL[5110 Salaries]
    OE --> RENT[5120 Rent]

    style A fill:#4ade80,color:#000
    style L fill:#f87171,color:#000
    style E fill:#60a5fa,color:#000
    style R fill:#a78bfa,color:#000
    style EX fill:#fb923c,color:#000
```

3. AccountType

Purpose: Defines account categories for double-entry bookkeeping.

Column	Type	Constraints	Description
id	INT	PK, AUTOINCREMENT	Primary key
name	VARCHAR(50)	UNIQUE, NOT NULL	Asset, Liability, Equity, Revenue, Expense
normalBalance	ENUM	NOT NULL	debit or credit
createdAt	TIMESTAMP	NOT NULL	Record creation

Enums:

```
enum NormalBalance {
  debit   // Asset, Expense accounts increase with debits
  credit  // Liability, Equity, Revenue accounts increase with credits
}
```

Seed data:

```
1 | Asset      | debit
2 | Liability  | credit
3 | Equity     | credit
4 | Revenue    | credit
5 | Expense    | debit
```

---

4. Account

Purpose: Chart of Accounts. Hierarchical GL accounts.

Column	Type	Constraints	Description
id	UUID	PK	Primary key
organizationId	UUID	FK → Organization, NOT NULL	Organization scope
code	VARCHAR(10)	NOT NULL	Account code (e.g., "1000", "4000")
name	VARCHAR(255)	NOT NULL	Account name (e.g., "Cash", "Revenue")

Column	Type	Constraints	Description
accountTypeId	INT	FK → AccountType, NOT NULL	Account category
currencyCode	CHAR(3)	NOT NULL, default 'EUR'	Account currency
parentAccountId	UUID	FK → Account, NULL	Parent account (for sub-accounts)
isActive	BOOLEAN	NOT NULL, default true	Active status
createdAt	TIMESTAMP	NOT NULL	Record creation
updatedAt	TIMESTAMP	NOT NULL	Last update

Indexes:

- Primary key: `id`
- Unique: `(organizationId, code)`
- Index: `idx_accounts_organization` on organizationId
- Index: `idx_accounts_type` on accountTypeId

Business rules:

- Code MUST be unique within organization
- Cannot delete account with transactions
- Parent-child hierarchy for sub-accounts (e.g., 1000 → 1001, 1002)

Contacts (Customers & Vendors)

5. Contact

Purpose: Customers (invoice recipients) and vendors (expense payees).

Column	Type	Constraints	Description
id	UUID	PK	Primary key
organizationId	UUID	FK → Organization, NOT NULL	Organization scope
type	ENUM	NOT NULL	customer, vendor, both
name	VARCHAR(255)	NOT NULL	Contact name
email	VARCHAR(255)	NULL	Email address
phone	VARCHAR(50)	NULL	Phone number

Column	Type	Constraints	Description
registrationNumber	VARCHAR(50)	NULL	Company registration number
vatNumber	VARCHAR(50)	NULL	VAT number
addressLine1	VARCHAR(255)	NULL	Street address
addressLine2	VARCHAR(255)	NULL	Apt/suite
city	VARCHAR(100)	NULL	City
postalCode	VARCHAR(20)	NULL	Postal/ZIP code
country	CHAR(2)	NULL	ISO 3166-1 alpha-2
currencyCode	CHAR(3)	NOT NULL, default 'EUR'	Preferred currency
paymentTerms	INT	NOT NULL, default 30	Payment terms in days
notes	TEXT	NULL	Free-text notes
isActive	BOOLEAN	NOT NULL, default true	Active status
createdAt	TIMESTAMP	NOT NULL	Record creation
updatedAt	TIMESTAMP	NOT NULL	Last update

Indexes:

- Primary key: `id`
- Index: `idx_contacts_organization` on organizationId
- Index: `idx_contacts_type` on type

Enums:

```
enum ContactType {
  customer  // Invoice recipient
  vendor    // Expense payee
  both      // Can be both customer and vendor
}
```

Business rules:

- Soft delete (isActive = false) if has invoices/expenses
- currencyCode determines default invoice/expense currency

Invoicing

6. Invoice

Purpose: Sales invoices (outgoing). Revenue recognition.

Column	Type	Constraints	Description
id	UUID	PK	Primary key
organizationId	UUID	FK → Organization, NOT NULL	Organization scope
customerId	UUID	FK → Contact, NOT NULL	Invoice recipient
invoiceNumber	VARCHAR(50)	NOT NULL	Auto-generated (e.g., INV-2026-001)
invoiceDate	DATE	NOT NULL	Invoice issue date
dueDate	DATE	NOT NULL	Payment due date
currencyCode	CHAR(3)	NOT NULL	Invoice currency
exchangeRate	DECIMAL(12,6)	NOT NULL, default 1.0	Exchange rate at invoiceDate
subtotal	DECIMAL(19,4)	NOT NULL	Sum of line totals (before tax)
taxAmount	DECIMAL(19,4)	NOT NULL, default 0	Total VAT/tax
discountAmount	DECIMAL(19,4)	NOT NULL, default 0	Total discount
totalAmount	DECIMAL(19,4)	NOT NULL	subtotal + taxAmount - discountAmount
baseAmount	DECIMAL(19,4)	NOT NULL	Converted to org baseCurrency
status	ENUM	NOT NULL, default 'draft'	Invoice status
sentAt	TIMESTAMP	NULL	When invoice was sent
viewedAt	TIMESTAMP	NULL	When customer viewed (email tracking)
paidAt	TIMESTAMP	NULL	When marked as paid
notes	TEXT	NULL	Internal notes
terms	TEXT	NULL	Payment terms text
pdfUrl	VARCHAR(500)	NULL	Cloudflare R2 URL
createdBy	UUID	FK → User, NULL	Creator user
createdAt	TIMESTAMP	NOT NULL	Record creation
updatedAt	TIMESTAMP	NOT NULL	Last update

Indexes:

- Primary key: `id`
- Unique: `(organizationId, invoiceNumber)`
- Index: `idx_invoices_organization` on organizationId
- Index: `idx_invoices_customer` on customerId
- Index: `idx_invoices_status` on status
- Index: `idx_invoices_due_date` on dueDate
- Composite: `idx_invoices_org_status_date` on (organizationId, status, invoiceDate)

Enums:

```
enum InvoiceStatus {
  draft      // Being edited
  sent       // Sent to customer
  viewed     // Customer viewed email
  paid       // Payment received
  overdue    // Past dueDate and unpaid
  cancelled  // Voided
}
```

Invoice Status Transitions:

```
stateDiagram-v2
    [*] --> draft : Created
    draft --> sent : Send email\n(creates GL Transaction)
    sent --> viewed : Tracking pixel loaded
    viewed --> paid : Mark as paid\n(creates GL Transaction)
    sent --> paid : Mark as paid\n(creates GL Transaction)
    draft --> cancelled : Cancel
    sent --> cancelled : Cancel\n(reverses Transaction)
    viewed --> cancelled : Cancel\n(reverses Transaction)
    paid --> [*]
    cancelled --> [*]

    note right of draft
        Can edit line items,\ndates, amounts
    end note
    note right of sent
        Locked — no edits\nPDF generated & stored in R2
    end note
    note right of overdue
        Automated check: dueDate < today\nAND status != paid
```

```
    end note
```

Business rules:

- invoiceNumber auto-generated on first save
- exchangeRate locked at invoiceDate (NEVER recalculate)
- baseAmount = totalAmount * exchangeRate
- Cannot edit invoice unless status = draft
- When status changes to 'paid', create Transaction (debit BankAccount, credit AccountsReceivable)

---

7. InvoiceItem

Purpose: Line items on invoices.

Column	Type	Constraints	Description
id	UUID	PK	Primary key
invoiceId	UUID	FK → Invoice, CASCADE, NOT NULL	Parent invoice
lineNumber	INT	NOT NULL	Line order (1, 2, 3...)
description	VARCHAR(500)	NOT NULL	Item description
quantity	DECIMAL(10,2)	NOT NULL	Quantity sold
unitPrice	DECIMAL(19,4)	NOT NULL	Price per unit
taxRate	DECIMAL(5,2)	NOT NULL, default 0	VAT rate (20 = 20%)
lineTotal	DECIMAL(19,4)	NOT NULL	quantity * unitPrice
accountId	UUID	FK → Account, NULL	Revenue account
createdAt	TIMESTAMP	NOT NULL	Record creation

Indexes:

- Primary key: `id`
- Index: `idx_invoice_items_invoice` on invoiceId

Business rules:

- lineTotal = quantity * unitPrice (calculated before save)
- Tax amount = lineTotal * (taxRate / 100)
- accountId determines which revenue account is credited

Expenses

8. Expense

Purpose: Purchase tracking (incoming). Expense recognition.

Column	Type	Constraints	Description
id	UUID	PK	Primary key
organizationId	UUID	FK → Organization, NOT NULL	Organization scope
vendorId	UUID	FK → Contact, NULL	Vendor (optional)
expenseNumber	VARCHAR(50)	NOT NULL	Auto-generated (e.g., EXP-2026-001)
expenseDate	DATE	NOT NULL	Expense date
currencyCode	CHAR(3)	NOT NULL	Expense currency
exchangeRate	DECIMAL(12,6)	NOT NULL, default 1.0	Exchange rate at expenseDate
amount	DECIMAL(19,4)	NOT NULL	Total expense amount
baseAmount	DECIMAL(19,4)	NOT NULL	Converted to org baseCurrency
taxAmount	DECIMAL(19,4)	NOT NULL, default 0	VAT amount
category	VARCHAR(100)	NOT NULL	Expense category
paymentMethod	VARCHAR(50)	NULL	cash, card, bank_transfer, etc.
accountId	UUID	FK → Account, NULL	Expense account
description	TEXT	NULL	Expense description
receiptUrl	VARCHAR(500)	NULL	Cloudflare R2 URL
status	ENUM	NOT NULL, default 'pending'	Approval status
approvedBy	UUID	FK → User, NULL	Approver user
approvedAt	TIMESTAMP	NULL	Approval timestamp
paidAt	TIMESTAMP	NULL	Payment timestamp
createdBy	UUID	FK → User, NULL	Creator user
createdAt	TIMESTAMP	NOT NULL	Record creation
updatedAt	TIMESTAMP	NOT NULL	Last update

Indexes:

- Primary key: `id`
- Unique: `(organizationId, expenseNumber)`
- Index: `idx_expenses_organization` on organizationId
- Index: `idx_expenses_vendor` on vendorId
- Index: `idx_expenses_category` on category
- Index: `idx_expenses_date` on expenseDate
- Composite: `idx_expenses_org_date_category` on (organizationId, expenseDate, category)

Enums:

```
enum ExpenseStatus {
  pending   // Awaiting approval
  approved  // Approved, ready to pay
  paid      // Payment made
  rejected  // Approval denied
}
```

Business rules:

- expenseNumber auto-generated
- exchangeRate locked at expenseDate
- baseAmount = amount * exchangeRate
- When status → approved, create Transaction (debit ExpenseAccount, credit AccountsPayable)
- When status → paid, create Transaction (debit AccountsPayable, credit BankAccount)

Transactions (Double-Entry Ledger)

9. Transaction

Purpose: General ledger transactions. Every financial event creates a transaction.

Column	Type	Constraints	Description
id	UUID	PK	Primary key
organizationId	UUID	FK → Organization, NOT NULL	Organization scope
transactionDate	DATE	NOT NULL	Transaction date
description	VARCHAR(255)	NOT NULL	Transaction description
debitAccountId	UUID	FK → Account, NOT NULL	Account to debit

Column	Type	Constraints	Description
creditAccountId	UUID	FK → Account, NOT NULL	Account to credit
amount	DECIMAL(19,4)	NOT NULL	Transaction amount
currencyCode	CHAR(3)	NOT NULL	Transaction currency
exchangeRate	DECIMAL(12,6)	NOT NULL, default 1.0	Exchange rate at transactionDate
baseAmount	DECIMAL(19,4)	NOT NULL	Converted to org baseCurrency
referenceType	VARCHAR(50)	NULL	invoice, expense, payment, manual
referenceId	UUID	NULL	Invoice/Expense ID
locked	BOOLEAN	NOT NULL, default false	Immutable if true
lockedAt	TIMESTAMP	NULL	When locked
reconciled	BOOLEAN	NOT NULL, default false	Matched to bank transaction
reconciledAt	TIMESTAMP	NULL	When reconciled
notes	TEXT	NULL	Free-text notes
createdBy	UUID	FK → User, NULL	Creator user
createdAt	TIMESTAMP	NOT NULL	Record creation

Indexes:

- Primary key: `id`
- Index: `idx_transactions_organization` on organizationId
- Index: `idx_transactions_date` on transactionDate
- Index: `idx_transactions_debit` on debitAccountId
- Index: `idx_transactions_credit` on creditAccountId
- Index: `idx_transactions_reference` on (referenceType, referenceId)
- Composite: `idx_transactions_org_date` on (organizationId, transactionDate)

Business rules:

- **DEBITS = CREDITS** — Every transaction has exactly one debit and one credit
- debitAccountId ≠ creditAccountId (enforced in API)
- Cannot edit if locked = true
- Cannot delete if reconciled = true
- exchangeRate locked at transactionDate
- baseAmount = amount * exchangeRate

Common transaction patterns:

1. **Invoice created (draft → sent):**
   - Debit: Accounts Receivable (Asset)
   - Credit: Revenue (Revenue)
2. **Invoice paid:**
   - Debit: Bank Account (Asset)
   - Credit: Accounts Receivable (Asset)
3. **Expense approved:**
   - Debit: Expense Account (Expense)
   - Credit: Accounts Payable (Liability)
4. **Expense paid:**
   - Debit: Accounts Payable (Liability)
   - Credit: Bank Account (Asset)

Banking & Reconciliation

10. BankAccount

Purpose: Bank account metadata.

Column	Type	Constraints	Description
id	UUID	PK	Primary key
organizationId	UUID	FK → Organization, NOT NULL	Organization scope
accountId	UUID	FK → Account, NOT NULL	GL account (must be Asset)
bankName	VARCHAR(255)	NOT NULL	Bank name
accountNumber	VARCHAR(50)	NULL	Account number
iban	VARCHAR(50)	NULL	IBAN
currencyCode	CHAR(3)	NOT NULL, default 'EUR'	Account currency
currentBalance	DECIMAL(19,4)	NOT NULL, default 0	Current balance
isActive	BOOLEAN	NOT NULL, default true	Active status
createdAt	TIMESTAMP	NOT NULL	Record creation
updatedAt	TIMESTAMP	NOT NULL	Last update

Indexes:

- Primary key: `id`
- Index: `idx_bank_accounts_organization` on organizationId

Business rules:

- accountId MUST be Asset type account
- currentBalance updated when transactions created
- Soft delete (isActive = false)

---

11. BankTransaction

Purpose: Bank statement imports. For reconciliation.

Column	Type	Constraints	Description
id	UUID	PK	Primary key
bankAccountId	UUID	FK → BankAccount, CASCADE, NOT NULL	Parent bank account
transactionDate	DATE	NOT NULL	Transaction date
amount	DECIMAL(19,4)	NOT NULL	Positive = credit, negative = debit
description	VARCHAR(500)	NULL	Bank description
reference	VARCHAR(255)	NULL	Reference number
reconciled	BOOLEAN	NOT NULL, default false	Matched to GL transaction
matchedTransactionId	UUID	NULL	GL transaction ID
createdAt	TIMESTAMP	NOT NULL	Record creation

Indexes:

- Primary key: `id`
- Index: `idx_bank_transactions_account` on bankAccountId
- Index: `idx_bank_transactions_date` on transactionDate

Business rules:

- Imported from CSV bank statements
- Matched to GL transactions via reconciliation workflow
- reconciled = true when matched

Multi-Currency

12. Currency

Purpose: Supported currencies.

Column	Type	Constraints	Description
code	CHAR(3)	PK	ISO 4217 currency code
name	VARCHAR(100)	NOT NULL	Currency name
symbol	VARCHAR(10)	NULL	Currency symbol
decimalPlaces	SMALLINT	NOT NULL, default 2	Decimal precision
isActive	BOOLEAN	NOT NULL, default true	Active status
createdAt	TIMESTAMP	NOT NULL	Record creation

Seed data:

```
EUR | Euro          | €    | 2
RSD | Serbian Dinar | din. | 2
BAM | Bosnian Mark  | KM   | 2
HRK | Croatian Kuna | kn   | 2
USD | US Dollar     | $    | 2
```

---

13. ExchangeRate

Purpose: Historical exchange rates.

Column	Type	Constraints	Description
id	UUID	PK	Primary key
baseCurrency	CHAR(3)	FK → Currency, NOT NULL	From currency
targetCurrency	CHAR(3)	FK → Currency, NOT NULL	To currency
rate	DECIMAL(12,6)	NOT NULL	Exchange rate
effectiveDate	DATE	NOT NULL	Rate effective date
source	VARCHAR(50)	NULL	ECB, fixer.io, manual
lastUpdated	TIMESTAMP	NOT NULL	Last update timestamp

Indexes:

- Primary key: `id`
- Unique: `(baseCurrency, targetCurrency, effectiveDate)`
- Index: `idx_exchange_rates_date` on effectiveDate
- Index: `idx_exchange_rates_pair` on (baseCurrency, targetCurrency)

Business rules:

- Rates fetched daily from ECB or fixer.io API
- When creating transaction, rate is locked at transaction date
- If no rate for exact date, use nearest available (warn in logs)

Audit Trail

14. LoggedAction

Purpose: Immutable audit log. Captures all INSERT/UPDATE/DELETE.

Column	Type	Constraints	Description
eventId	BIGINT	PK, AUTOINCREMENT	Event ID
schemaName	TEXT	NOT NULL	Database schema (default: public)
tableName	TEXT	NOT NULL	Table name
userId	UUID	FK → User, NULL	User who performed action
actionTimestamp	TIMESTAMP	NOT NULL, default now()	When action occurred
action	ENUM	NOT NULL	INSERT, UPDATE, DELETE
rowData	JSONB	NULL	Full row data before change
changedFields	JSONB	NULL	Changed fields (UPDATE only)
queryText	TEXT	NULL	SQL query (if available)
clientIp	INET	NULL	Client IP address
applicationName	TEXT	NOT NULL, default 'fiken-clone-api'	Application identifier

Indexes:

- Primary key: `eventId`
- Index: `idx_logged_actions_timestamp` on actionTimestamp
- Index: `idx_logged_actions_table` on tableName
- Index: `idx_logged_actions_user` on userId

Enums:

```
enum AuditAction {
  INSERT
  UPDATE
  DELETE
}
```

Business rules:

- **APPEND-ONLY** — NEVER delete or update records
- Triggered via Prisma middleware (automatic)
- Used for: compliance, debugging, rollback simulation

Schema Version

15. SchemaVersion

Purpose: Migration tracking.

Column	Type	Constraints	Description
version	VARCHAR(20)	PK	Version string (e.g., "1.0.0")
appliedAt	TIMESTAMP	NOT NULL, default now()	Migration timestamp
description	TEXT	NULL	Migration description

Business rules:

- Updated by Prisma migrations
- Used to verify schema version matches application version

Data Types & Precision

NUMERIC(19,4) for ALL Money

CRITICAL: NEVER use `float`, `double`, or JavaScript `number` for currency.

- Precision: 19 digits total, 4 decimal places
- Range: -999,999,999,999,999.9999 to +999,999,999,999,999.9999
- Prisma type: `Decimal`

- PostgreSQL type: `NUMERIC(19,4)`

Why:

- JavaScript `number` has 53-bit precision (safe up to 2^53 - 1 = 9,007,199,254,740,991)
- Financial calculations require exact decimal precision
- Example: 0.1 + 0.2 = 0.30000000000000004 (float error)

Usage in code:

```
import { Decimal } from '@prisma/client/runtime'


const amount = new Decimal('125000.0000')
const taxRate = new Decimal('0.20')
const taxAmount = amount.times(taxRate)  // 25000.0000
```

Constraints Summary

Primary Keys

All tables use UUID primary keys (except AccountType uses INT auto-increment).

Foreign Keys

All foreign keys have `onDelete: Cascade` (deleting organization deletes all data).

Unique Constraints

- User.email
- Account.(organizationId, code)
- Invoice.(organizationId, invoiceNumber)
- Expense.(organizationId, expenseNumber)
- ExchangeRate.(baseCurrency, targetCurrency, effectiveDate)

Check Constraints

(Enforced in API layer, not database):

- amount > 0 for all financial amounts
- dueDate >= invoiceDate for invoices

- debitAccountId ≠ creditAccountId for transactions

Indexes Strategy

Query Patterns Optimized

1. **List by organization + filter:**
   - `(organizationId, status, date)` composite index on invoices
   - `(organizationId, category, date)` composite index on expenses
2. **Foreign key lookups:**
   - All foreign keys have indexes
3. **Date range queries:**
   - Dedicated indexes on `transactionDate`, `invoiceDate`, `expenseDate`, `dueDate`
4. **Reconciliation:**
   - Index on `(referenceType, referenceId)` for transaction lookups

Migration Commands

```
# Generate Prisma Client
npx prisma generate

# Create migration
npx prisma migrate dev --name migration_name

# Apply migrations (production)
npx prisma migrate deploy

# Reset database (dev only)
npx prisma migrate reset

# Seed initial data
npx prisma db seed
```

Seed Data

AccountType

```
INSERT INTO account_types (id, name, normal_balance) VALUES
(1, 'Asset', 'debit'),
(2, 'Liability', 'credit'),
(3, 'Equity', 'credit'),
(4, 'Revenue', 'credit'),
(5, 'Expense', 'debit');
```

Currency

```
INSERT INTO currencies (code, name, symbol, decimal_places) VALUES
('EUR', 'Euro', '€', 2),
('RSD', 'Serbian Dinar', 'din.', 2),
('BAM', 'Bosnian Mark', 'KM', 2),
('HRK', 'Croatian Kuna', 'kn', 2),
('USD', 'US Dollar', '$', 2);
```

End of Database Schema

Revision #4
Created 2026-02-23 10:47:52 UTC by John
Updated 2026-07-05 20:02:22 UTC by John