

# Database — Schema & Models

## Bilko Database — Prisma + PostgreSQL

### BookStack — Provjeri PRVO

Prije traženja bilo čega — provjeri BookStack (<https://docs.basicconsulting.no>). Centralna baza znanja za tools, skills, hooks, agents, rules, projekte, klijente, dokumentaciju. Ako odgovor postoji tamo — NE TRAŽI dalje.

## Schema Location

`prisma/schema.prisma` — 15 models, fully defined

## Database Models (15)

### Core:

- Organization — Multi-tenant root (baseCurrency, country, language)
- User — RBAC (owner, admin, accountant, viewer)

### Chart of Accounts:

- AccountType — Asset, Liability, Equity, Revenue, Expense
- Account — Hierarchical CoA with parent-child relations

### Contacts:

- Contact — Customers, vendors, or both (type enum)

### Invoicing:

- Invoice — Sales invoices with multi-currency support

- InvoiceItem — Line items with tax rates

### Expenses:

- Expense — Purchase tracking with approval workflow

### Transactions:

- Transaction — Double-entry ledger (debit + credit accounts)

### Banking:

- BankAccount — Bank account metadata
- BankTransaction — Bank statement imports for reconciliation

### Multi-Currency:

- Currency — Currency definitions (EUR, RSD, BAM, HRK, etc.)
- ExchangeRate — Historical exchange rates by date

### Audit:

- LoggedAction — Immutable audit trail (APPEND-ONLY)
- SchemaVersion — Migration tracking

# Key Design Decisions

## 1. NUMERIC(19,4) for Money

**NEVER use float or JavaScript number for currency.**

- Prisma type: `Decimal` (maps to PostgreSQL NUMERIC)
- Precision: 19 digits total, 4 decimal places
- Range: -999,999,999,999,999.9999 to +999,999,999,999,999.9999

## 2. Double-Entry Bookkeeping

Every financial event creates a `Transaction` with:

- `debitAccountId` — Account to debit
- `creditAccountId` — Account to credit
- `amount` — MUST be equal for both sides
- Balance = sum(debits) - sum(credits) per account

## 3. Multi-Currency with Rate Locking

- `Invoice.exchangeRate` — Locked at invoice date
- `Transaction.exchangeRate` — Locked at transaction date
- `baseAmount` — Amount converted to org's baseCurrency
- **NEVER recalculate** historical transactions with current rates

## 4. Immutable Audit Trail

`LoggedAction` table:

- **APPEND-ONLY** — NEVER delete or update
- Captures: table name, user ID, action (INSERT/UPDATE/DELETE), old/new values
- Used for: compliance, debugging, rollback simulation

## 5. Transaction Locking

- `Transaction.locked` — Once true, record is immutable
- Locked transactions cannot be edited or deleted
- Used for: end-of-period close, tax reporting

## 6. Organization-Scoped Multi-Tenancy

- Every record has `organizationId` foreign key
- Queries MUST filter by org (enforced in API middleware)
- No cross-org data access

## 7. UUID Primary Keys

- All IDs: `uuid_generate_v4()` (PostgreSQL function)
- NEVER use auto-increment for business data
- Portable across systems, no collisions

# Migration Rules

1. **Never edit existing migrations** — Always create new ones
2. **Test migrations on copy** — Never run on production first
3. **Backward compatible** — Additive changes only
4. **Data migrations separate** — Use Prisma seed or custom scripts
5. **Rollback plan** — Document how to undo breaking changes

# Naming Conventions

- **DB columns:** snake\_case (via `@map`)
- **Prisma fields:** camelCase
- **Indexes:** `idx_{table}_{column(s)}`
- **Foreign keys:** Auto-generated by Prisma

# Indexes

Defined for:

- All foreign keys (automatic)
- Common query patterns (org + date, org + status)
- Unique constraints (org + code, org + invoice number)

# Enums

- UserRole: owner, admin, accountant, viewer
- NormalBalance: debit, credit
- ContactType: customer, vendor, both
- InvoiceStatus: draft, sent, viewed, paid, overdue, cancelled
- ExpenseStatus: pending, approved, paid, rejected
- AuditAction: INSERT, UPDATE, DELETE

# Development 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
```

```
# Open Prisma Studio
```

```
npx prisma studio
```

# Critical Rules

1. **NUMERIC for money** — NEVER float
2. **Double-entry enforced** — Every transaction has debit + credit
3. **Exchange rates locked** — At transaction date, NEVER recalculate
4. **Audit is append-only** — NEVER delete LoggedAction records
5. **UUID everywhere** — NEVER expose auto-increment IDs

---

Revision #8

Created 2026-02-24 10:11:13 UTC by John

Updated 2026-07-12 20:00:50 UTC by John