Migration: Add Margin System and Invoice Updates

Date: December 10, 2024
Branch: expenses-structure
Migration Name: add-margin-system-and-invoice-updates

Overview

This migration implements a comprehensive margin tracking system and enhances the invoice model with sender/receiver fields, payment model enum, and agency payment tracking capabilities.

Schema Changes

1. New Enums

PaymentModel

```
enum PaymentModel {
  GROSS
  PAYROLL
  PAYROLL_WE_PAY
  SPLIT
}
```

MarginType

```
enum MarginType {
  FIXED
  VARIABLE
  CUSTOM
}
```

2. New Table: Margin

A dedicated table to track margins separately from invoices, supporting:
- Fixed/Variable/Custom margin types
- Percentage-based and fixed amount margins
- Override tracking (who, when, why)
- Complete audit trail

Fields:
- `id` (String, cuid, primary key)
- `invoiceId` (String, unique) - 1-to-1 relation with Invoice
- `contractId` (String) - Many-to-1 relation with Contract
- `marginType` (MarginType enum)
- `marginPercentage` (Decimal, optional) - For percentage-based margins
- `marginAmount` (Decimal, optional) - For fixed amount margins
- `calculatedMargin` (Decimal) - The final calculated margin value

- `isOverridden` (Boolean, default false)
- `overriddenBy` (String, optional) - userId who made override
- `overriddenAt` (DateTime, optional)
- `notes` (String, optional)
- `createdAt`, `updatedAt` (DateTime)

Indexes:
- `invoiceId` (for 1-to-1 lookups)
- `contractId` (for contract-based queries)
- `overriddenBy` (for audit queries)

3. Invoice Model Updates

New Fields:
- `senderId` (String, optional) - Reference to User (invoice sender)
- `receiverId` (String, optional) - Reference to User (invoice receiver)
- `paymentModel` (PaymentModel enum, optional) - Inherited from contract
- `agencyMarkedPaidAt` (DateTime, optional) - When agency marked invoice as paid
- `agencyMarkedPaidBy` (String, optional) - userId who marked as paid
- `paymentReceivedAt` (DateTime, optional) - When payment was actually received
- `paymentReceivedBy` (String, optional) - userId who confirmed receipt

New Relations:
- `sender` → User (via senderId)
- `receiver` → User (via receiverId)
- `margin` → Margin (1-to-1, optional)
- `agencyMarkedPaidByUser` → User (via agencyMarkedPaidBy)
- `paymentReceivedByUser` → User (via paymentReceivedBy)

New Indexes:
- `senderId`
- `receiverId`
- `agencyMarkedPaidBy`
- `paymentReceivedBy`

4. Contract Model Updates

New Fields:
- `paymentModel` (PaymentModel enum, optional) - Default payment model for this contract

New Relations:
- `margins` → Margin[] (one-to-many)

5. User Model Updates

New Relations:
- `invoicesSent` → Invoice[] (via InvoiceSender relation)
- `invoicesReceived` → Invoice[] (via InvoiceReceiver relation)
- `invoicesAgencyMarkedPaid` → Invoice[] (via InvoiceAgencyMarkedPaidBy relation)
- `invoicesPaymentReceived` → Invoice[] (via InvoicePaymentReceivedBy relation)
- `marginsOverridden` → Margin[] (via MarginOverriddenBy relation)

Expected SQL Migration

When you run `npx prisma migrate dev --name add-margin-system-and-invoice-updates`, Prisma will generate SQL similar to:

```sql
-- CreateEnum
CREATE TYPE "PaymentModel" AS ENUM ('GROSS', 'PAYROLL', 'PAYROLL_WE_PAY', 'SPLIT');

-- CreateEnum
CREATE TYPE "MarginType" AS ENUM ('FIXED', 'VARIABLE', 'CUSTOM');

-- AlterTable: Add new columns to invoices
ALTER TABLE "invoices"
ADD COLUMN "senderId" TEXT,
ADD COLUMN "receiverId" TEXT,
ADD COLUMN "paymentModel" "PaymentModel",
ADD COLUMN "agencyMarkedPaidAt" TIMESTAMP(3),
ADD COLUMN "agencyMarkedPaidBy" TEXT,
ADD COLUMN "paymentReceivedAt" TIMESTAMP(3),
ADD COLUMN "paymentReceivedBy" TEXT;

-- AlterTable: Add new column to contracts
ALTER TABLE "contracts"
ADD COLUMN "paymentModel" "PaymentModel";

-- CreateTable: margins
CREATE TABLE "margins" (
    "id" TEXT NOT NULL,
    "invoiceId" TEXT NOT NULL,
    "contractId" TEXT NOT NULL,
    "marginType" "MarginType" NOT NULL,
    "marginPercentage" DECIMAL(5,2),
    "marginAmount" DECIMAL(10,2),
    "calculatedMargin" DECIMAL(10,2) NOT NULL,
    "isOverridden" BOOLEAN NOT NULL DEFAULT false,
    "overriddenBy" TEXT,
    "overriddenAt" TIMESTAMP(3),
    "notes" TEXT,
    "createdAt" TIMESTAMP(3) NOT NULL DEFAULT CURRENT_TIMESTAMP,
    "updatedAt" TIMESTAMP(3) NOT NULL,

    CONSTRAINT "margins_pkey" PRIMARY KEY ("id")
);

-- CreateIndex
CREATE UNIQUE INDEX "margins_invoiceId_key" ON "margins"("invoiceId");

-- CreateIndex
CREATE INDEX "margins_invoiceId_idx" ON "margins"("invoiceId");

-- CreateIndex
CREATE INDEX "margins_contractId_idx" ON "margins"("contractId");

-- CreateIndex
CREATE INDEX "margins_overriddenBy_idx" ON "margins"("overriddenBy");

-- CreateIndex
CREATE INDEX "invoices_senderId_idx" ON "invoices"("senderId");

-- CreateIndex
CREATE INDEX "invoices_receiverId_idx" ON "invoices"("receiverId");

-- CreateIndex
CREATE INDEX "invoices_agencyMarkedPaidBy_idx" ON "invoices"("agencyMarkedPaidBy");

-- CreateIndex
CREATE INDEX "invoices_paymentReceivedBy_idx" ON "invoices"("paymentReceivedBy");
```

```
-- AddForeignKey
ALTER TABLE "margins" ADD CONSTRAINT "margins_invoiceId_fkey"
    FOREIGN KEY ("invoiceId") REFERENCES "invoices"("id") ON DELETE CASCADE ON UPDATE CASCADE;

-- AddForeignKey
ALTER TABLE "margins" ADD CONSTRAINT "margins_contractId_fkey"
    FOREIGN KEY ("contractId") REFERENCES "contracts"("id") ON DELETE RESTRICT ON UPDATE CASCADE;

-- AddForeignKey
ALTER TABLE "margins" ADD CONSTRAINT "margins_overriddenBy_fkey"
    FOREIGN KEY ("overriddenBy") REFERENCES "users"("id") ON DELETE SET NULL ON UPDATE CASCADE;

-- AddForeignKey
ALTER TABLE "invoices" ADD CONSTRAINT "invoices_senderId_fkey"
    FOREIGN KEY ("senderId") REFERENCES "users"("id") ON DELETE SET NULL ON UPDATE CASCADE;

-- AddForeignKey
ALTER TABLE "invoices" ADD CONSTRAINT "invoices_receiverId_fkey"
    FOREIGN KEY ("receiverId") REFERENCES "users"("id") ON DELETE SET NULL ON UPDATE CASCADE;

-- AddForeignKey
ALTER TABLE "invoices" ADD CONSTRAINT "invoices_agencyMarkedPaidBy_fkey"
    FOREIGN KEY ("agencyMarkedPaidBy") REFERENCES "users"("id") ON DELETE SET NULL ON UPDATE CASCADE;

-- AddForeignKey
ALTER TABLE "invoices" ADD CONSTRAINT "invoices_paymentReceivedBy_fkey"
    FOREIGN KEY ("paymentReceivedBy") REFERENCES "users"("id") ON DELETE SET NULL ON UPDATE CASCADE;
```

How to Run This Migration

Option 1: Using Prisma Migrate (Recommended for Production)

When you have access to your database:

```
# Run the migration (creates migration file and applies it)
npx prisma migrate dev --name add-margin-system-and-invoice-updates

# Or if migration file already exists, just apply pending migrations
npx prisma migrate deploy
```

Option 2: Using Prisma DB Push (Development)

For rapid development without migration files:

```
npx prisma db push
```

Option 3: Manual SQL Execution

If you prefer to run SQL manually:

1. Copy the SQL from the “Expected SQL Migration” section above
2. Connect to your PostgreSQL database
3. Execute the SQL in a transaction
4. Run `npx prisma generate` to update Prisma Client

Validation Steps

After running the migration:

1. **Verify Schema**:
   ```
   bash
      npx prisma validate
   ```

2. **Regenerate Prisma Client**:
   ```
   bash
      npx prisma generate
   ```

3. **Check Database**:
   ```
   bash
      npx prisma studio
   ```
   - Verify `margins` table exists
   - Verify new columns in `invoices` table
   - Verify new enum types (`PaymentModel` , `MarginType`)

Breaking Changes

⚠️ **None** - This migration is additive only:
- All new fields are optional
- No existing data will be modified
- No columns are being removed or renamed

Rollback Procedure

If you need to rollback this migration:

```sql
-- Drop foreign keys first
ALTER TABLE "invoices" DROP CONSTRAINT IF EXISTS "invoices_senderId_fkey";
ALTER TABLE "invoices" DROP CONSTRAINT IF EXISTS "invoices_receiverId_fkey";
ALTER TABLE "invoices" DROP CONSTRAINT IF EXISTS "invoices_agencyMarkedPaidBy_fkey";
ALTER TABLE "invoices" DROP CONSTRAINT IF EXISTS "invoices_paymentReceivedBy_fkey";
ALTER TABLE "margins" DROP CONSTRAINT IF EXISTS "margins_invoiceId_fkey";
ALTER TABLE "margins" DROP CONSTRAINT IF EXISTS "margins_contractId_fkey";
ALTER TABLE "margins" DROP CONSTRAINT IF EXISTS "margins_overriddenBy_fkey";

-- Drop table
DROP TABLE IF EXISTS "margins";

-- Drop invoice columns
ALTER TABLE "invoices"
DROP COLUMN IF EXISTS "senderId",
DROP COLUMN IF EXISTS "receiverId",
DROP COLUMN IF EXISTS "paymentModel",
DROP COLUMN IF EXISTS "agencyMarkedPaidAt",
DROP COLUMN IF EXISTS "agencyMarkedPaidBy",
DROP COLUMN IF EXISTS "paymentReceivedAt",
DROP COLUMN IF EXISTS "paymentReceivedBy";

-- Drop contract column
ALTER TABLE "contracts"
DROP COLUMN IF EXISTS "paymentModel";

-- Drop enums
DROP TYPE IF EXISTS "MarginType";
DROP TYPE IF EXISTS "PaymentModel";
```

Testing Recommendations

After migration, test:

1. **Margin Creation**: Create a margin record for an invoice
2. **Invoice Creation**: Create an invoice with sender/receiver
3. **Payment Tracking**: Test agency payment marking workflow
4. **Contract Updates**: Test contract payment model settings
5. **Queries**: Test all new indexed fields for performance

Related Files

- Schema: `prisma/schema.prisma`
- Margin Model: Lines 714-743
- Invoice Model: Lines 620-717
- Contract Model: Lines 409-518
- User Model: Lines 229-340

Additional Notes

- The `marginType` field in Contract remains as String for backward compatibility. Consider migrating to MarginType enum in a future migration if needed.
- All foreign keys use appropriate cascade rules:

- Margin → Invoice: CASCADE (margin deleted when invoice deleted)
- Margin → Contract: RESTRICT (prevent contract deletion if margins exist)
- User relations: SET NULL (preserve records if user deleted)