**Lista de entidades para Merkadit**

**Tablas de addresses**

**Countries**

* countryID: TINYINT AUTO\_INCREMENT PK
* name: VARCHAR(60)

**States**

* stateID: INT AUTO\_INCREMENT PK
* name: VARCHAR(60)
* countryID: TINYINT FK → Countries

**Cities**

* cityID: INT AUTO\_INCREMENT PK
* name: VARCHAR(60)
* stateID: INT FK → States

**Addresses**

* addressID: INT AUTO\_INCREMENT PK
* line1: VARCHAR(200)
* line2: VARCHAR(200)
* zipCode: VARCHAR(10)
* location: POINT
* cityID: INT FK → Cities

**AddressXUsers**

* addressID: INT PK, FK → Addresses
* userID: INT PK, FK → Users
* postTime: DATETIME DEFAULT NOW()
* enabled: BIT DEFAULT 1
* isDefault: BIT DEFAULT 0

**Currencies**

* currencyID
* name
* isoCode: VARCHAR(3)
* currencySymbol: VARCHAR(5)
* countryID

**ExchangeRates**

* exchangeRateID
* startDate: Datetime
* endDate: Datetime NULL
* exchangeRate: Decimal(10,4)
* currencySourceID: FK
* currencyDestinyID: FK
* current: BIT

**Tablas de usuario**

**Users**

* userID: INT AUTO\_INCREMENT PK
* email: VARCHAR(100) UNIQUE
* firstName: VARCHAR(50)
* lastName: VARCHAR(50)
* passwordHash: VARBINARY(250)
* createdDate: DATETIME DEFAULT NOW()
* lastLogin: DATETIME
* isActive: BIT DEFAULT 1
* checksum: VARBINARY(250)

**UserXContacts**

* contactID
* userID
* ContactType FK
* value: varchar(80)
* postTime
* enabled
* deleted

**ContactTypes**

* contactTypeID
* name

**UserXRoles**

* userID: INT PK, FK → Users
* roleID: INT PK, FK → Roles
* postTime: DATETIME DEFAULT NOW()
* enabled: BIT DEFAULT 1
* checksum: VARBINARY(250)

**Roles**

* roleID: INT AUTO\_INCREMENT PK
* name: VARCHAR(30)
* description: VARCHAR(200)

**PermissionXRoles**

* permissionID: INT PK, FK → Permissions
* roleID: INT PK, FK → Roles
* postTime: DATETIME DEFAULT NOW()
* enabled: BIT DEFAULT 1
* checksum: VARBINARY(250)

**Permissions**

* permissionID: INT AUTO\_INCREMENT PK
* name: VARCHAR(50)
* description: VARCHAR(200)
* code: VARCHAR(20) UNIQUE
* module: VARCHAR(50)

**Tablas de comercios**

**Commerces**

* commerceID: INT AUTO\_INCREMENT PK
* name: VARCHAR(100)
* legalName: VARCHAR(200)
* taxID: VARCHAR(30) UNIQUE
* businessType: VARCHAR(50)
* ownerUserID: INT FK → Users
* createdDate: DATETIME DEFAULT NOW()
* isActive: BIT DEFAULT 1

**CommerceXContacts**

* contactID
* commerceID
* contactTypeID FK
* value: varchar(80)
* postTime
* enabled
* deleted

**Buildings**

Building registered in the app

* buildingID: INT AUTO\_INCREMENT PK
* name: VARCHAR(100)
* totalArea: DECIMAL(10,2)
* floors: TINYINT
* openingTime: TIME
* closingTime: TIME
* adminUserID: INT FK → Users
* addressID: INT FK → Addresses
* initialInvestment: DECIMAL(15,2)
* createdDate: DATETIME DEFAULT NOW()

**Floors**

* buildingID
* name
* floorNumber

**SpaceTypes**

If the space is designed for a restaurant, a kiosk

* spaceTypeID: TINYINT AUTO\_INCREMENT PK
* name: VARCHAR(50)
* description: VARCHAR(200)

**Spaces**

The concrete space that you can rent

* spaceID: INT AUTO\_INCREMENT PK
* floorID: INT FK → floors
* spaceCode: VARCHAR(20)
* name: VARCHAR(100)
* area: DECIMAL(10,2)
* spaceTypeID: TINYINT FK → SpaceTypes
* spaceStatusID FK
* baseRent: DECIMAL(12,2)

**SpaceStatus**

* statusID
* name

**Tablas de contratos**

**Contracts**

* contractID: INT AUTO\_INCREMENT PK
* contractNumber: VARCHAR(50) UNIQUE
* commerceID: INT FK → Commerces
* spaceID: INT FK → Spaces
* startDate: DATE
* endDate: DATE
* document: varchar(500)
* baseRent: DECIMAL(12,2)
* currencyID
* commissionPercentage: DECIMAL(5,2) DEFAULT 0
* scheduleID FK
* contractStatusID FK
* createdDate: DATETIME DEFAULT NOW()
* createdBy: userID FK

**Schedules**

* scheduleID
* scheduleRecurrencyID
* startDate: Datetime
* endDate: Datetime
* lastExecute: Datetime NULL
* nextExecute: Datetime
* enabled: BIT
* deleted: BIT

**ScheduleRecurrencies**

Says how frequent the schedule is. Monthly, daily, annual…

* scheduleRecurrencyID
* name
* intervalDays: int

**contractStatus**

* contractStatusID
* name

**Settlements**

* settlementID: INT AUTO\_INCREMENT PK
* contractID: INT FK → Contracts
* scheduleID:
* baseRentAmount: DECIMAL(12,2)
* totalSales: DECIMAL(12,2)
* commissionAmount: DECIMAL(12,2)
* totalAmount: DECIMAL(12,2)
* currencyID
* settlementDate: DATETIME DEFAULT NOW()
* settlementStatus FK
* paidDate: DATETIME
* createdBy: userID

**SettlementStatus** ENUM('pending', 'paid', 'overdue', 'cancelled')

* settlementStatusID
* name

**Tablas de inventario**

**Categories**

* categoryID: INT AUTO\_INCREMENT PK
* name: VARCHAR(50)
* description: VARCHAR(200)

**Products**

sku es un código único que se le asigna al producto. min stock es la minima cantidad de producto antes de ordenar. max stock es el nivel más alto de mantener en el inventario.

* productID: INT AUTO\_INCREMENT PK
* commerceID: INT FK → Commerces
* sku: VARCHAR(50)
* barcode: VARCHAR(50)
* name: VARCHAR(100)
* description: TEXT
* categoryID: INT FK → Categories
* price: DECIMAL(12,2)
* currencyID: INT FK → Currencies
* cost: DECIMAL(12,2)
* stockQuantity: INT DEFAULT 0
* minStock: INT DEFAULT 0
* maxStock: INT
* isActive: BIT DEFAULT 1
* createdDate: DATETIME DEFAULT NOW()
* deleted: BIT DEFAULT 0

**InventoryMovements**

Reference es la causa del movimiento

* movementID: INT AUTO\_INCREMENT PK
* productID: INT FK → Products
* movementType: ENUM('IN', 'OUT', 'ADJUSTMENT', 'RETURN')
* stockQuantity: INT
* referenceDescription: VARCHAR(100)
* referenceID: INT
* movementDate: DATETIME DEFAULT NOW()
* createdBy: VARCHAR(50)
* checksum: VARBINARY

**MovementTypes** ENUM('IN', 'OUT', 'ADJUSTMENT', 'RETURN')

* movementTypeID
* name

**Tablas de ventas**

**Customers**

* customerID: INT AUTO\_INCREMENT PK
* customerType: INT FK
* name: VARCHAR(50)
* taxID: VARCHAR(30)
* birthdate: Datetime
* addressID: FK
* createdDate: DATETIME DEFAULT NOW()
* enabled: BIT DEFAULT 1
* deleted: BIT DEFAULT 0

**CustomerTypes** ENUM(Company, person)

* customerTypeID
* name

**PaymentMethods**

* paymentMethodID: TINYINT AUTO\_INCREMENT PK
* name: VARCHAR(50)
* code: VARCHAR(20) UNIQUE
* requiresReference: BIT DEFAULT 0

**Payment**

* paymentID
* paymentMethodID
* paymentTypeID
* transactionAmount: Decimal(16,2)
* currencyID FK
* description:
* paymentDate: Datetime
* paymentReference: VARCHAR(100)
* paymentConfirmation: VARCHAR(100)
* paymentDate
* status:
* checksum

**PaymentTypes**

* paymentTypeID
* name
* description

**Sales**

* saleID: INT AUTO\_INCREMENT PK
* commerceID: INT FK → Commerces
* invoiceNumber: VARCHAR(50)
* saleDate: DATETIME DEFAULT NOW()
* customerID: INT FK → Customers
* subtotal: DECIMAL(12,2)
* discountAmount: DECIMAL(12,2) DEFAULT 0
* taxAmount: DECIMAL(12,2) DEFAULT 0
* totalAmount: DECIMAL(12,2)
* currencyID: FK
* computer: VARCHAR(50)
* userID: FK
* paymentMethodID: TINYINT FK → PaymentMethods
* paymentReference: VARCHAR(100)
* paymentConfirmation: VARCHAR(100)
* status: ENUM('completed', 'cancelled', 'pending', 'refunded')

**SaleDetails**

* saleDetailID: INT AUTO\_INCREMENT PK
* saleID: INT FK → Sales
* productID: INT FK → Products
* quantity: INT
* unitPrice: DECIMAL(12,2)
* discountAmount: DECIMAL(12,2) DEFAULT 0
* subtotal: DECIMAL(12,2)
* currencyID: FK

**Tablas Financieras**

**Investments**

* investmentID: INT AUTO\_INCREMENT PK
* spaceID: INT FK → spaces
* contractID: INT FK → contratos
* description: VARCHAR(500)
* amount: DECIMAL(15,2)
* currencyID: FK INT
* investmentDate: DATE
* investmentCategoryID: Tinyint FK
* userID: FK
* createdDate: DATETIME DEFAULT NOW()
* checksum: VARBINARY

**InvestmentCategories**

* investmentCategoryID TINYINT AUTO\_INCREMENT PK
* name : VARCHAR(50)
* description : VARCHAR(200)
* isCapital : BIT
* depreciationMonths : Smallint

**Expenses**

* expenseID: INT AUTO\_INCREMENT PK
* buildingID: INT FK → Buildings
* description: VARCHAR(500)
* amount: DECIMAL(12,2)
* expenseDate: DATE
* category: ENUM('utilities', 'security', 'cleaning', 'marketing', 'maintenance', 'other')
* invoiceNumber: VARCHAR(50)
* supplierName: VARCHAR(100)
* createdBy: VARCHAR(50)
* createdDate: DATETIME DEFAULT NOW()

**FinancialReports**

* reportID: INT AUTO\_INCREMENT PK
* spaceID: INT FK
* totalRevenue: Decimal(16,2)
* toalExpenses : Decimal(16,2)
* netIncome: Decimal(16,2)
* groossMargin :
* occupancyRate:
* currencyID
* documentURL: VARCHAR(250) NULL
* posttime: Datetime
* userID: Datetime

**Tablas de Logs**

**Logs**

* logID: BIGINT AUTO\_INCREMENT PK
* postTime: DATETIME DEFAULT NOW()
* description: VARCHAR(500)
* computer: VARCHAR(100)
* username: VARCHAR(50)
* ref1ID: INT
* ref2ID: INT
* value1: VARCHAR(200)
* value2: VARCHAR(200)
* logTypeID: TINYINT FK → LogTypes
* logLevelID: TINYINT FK → LogLevels
* logSourceID: TINYINT FK → LogSources
* checksum: VARBINARY(250)

**LogTypes**

* logTypeID: TINYINT AUTO\_INCREMENT PK
* name: VARCHAR(50)

**LogLevels**

* logLevelID: TINYINT AUTO\_INCREMENT PK
* name: VARCHAR(30)

**LogSources**

* logSourceID: TINYINT AUTO\_INCREMENT PK
* name: VARCHAR(50)