DropZero - Schema Database MongoDB Completo (v2.1)

1. Panoramica Architettura Database

Il database MongoDB di DropZero è organizzato in **7 collezioni principali** progettate per supportare:

- Gestione utenti (privati e amministratori)
- Associazione sensori/contatori agli utenti
- Letture consumi **settimanali** (figurative/da sensori)
- Storico consolidato consumi per grafici e analisi
- Tariffe e costi generali
- Generazione bollette e costi
- Dati aggregati territoriali anonimizzati

Frequenza letture: Settimanale (non giornaliera/oraria)
Periodo fatturazione: Mensile o semestrale
Conservazione dati: 7 anni (GDPR compliance)
MongoDB versione minima: 5.0+

---

2. Collezioni e Schema Dettagliato

2.1 COLLEZIONE: `users`

Descrizione: Gestione utenti (privati e amministratori)

Schema:
{
"_id": ObjectId,
"email": String, // Univoco, case-insensitive
"password": String, // Hash bcrypt (min 8 char, maiusc, numero, speciale)
"userType": String, // "PRIVATE" | "ADMIN"
"firstName": String, // Obbligatorio per privati
"lastName": String, // Obbligatorio per privati
"codiceFiscale": String, // Univoco per privati
"address": String, // Indirizzo completo
"phoneNumber": String, // Opzionale, internazionale
"verified": Boolean, // Email verificata?
"status": String, // "ACTIVE" | "SUSPENDED" | "DELETED"
"roles": [String], // ["user"] | ["admin"]
"lastLogin": ISODate, // Ultimo accesso
"lastLoginIP": String, // IP ultimo accesso
"createdAt": ISODate, // Data registrazione
"updatedAt": ISODate, // Ultimo aggiornamento
"preferences": {

"notificationChannels": [String], // ["PUSH", "EMAIL", "IN_APP"]
"language": String, // "it" | "en"
"theme": String // "LIGHT" | "DARK" | "AUTO"
}
}

Indici:
db.users.createIndex({ "email": 1 }, { unique: true })
db.users.createIndex({ "codiceFiscale": 1 }, { unique: true, sparse: true })
db.users.createIndex({ "userType": 1 })
db.users.createIndex({ "status": 1 })
db.users.createIndex({ "createdAt": -1 })

Esempio documento:
{
"_id": ObjectId("507f1f77bcf86cd799439011"),
"email": "mario.rossi@email.com",
"password": "$2b10...",
"userType": "PRIVATE",
"firstName": "Mario",
"lastName": "Rossi",
"codiceFiscale": "RSSMRA90M01H501U",
"address": "Via Roma 15, 40126 Bologna",
"phoneNumber": "+39 333 1234567",
"verified": true,
"status": "ACTIVE",
"roles": ["user"],
"lastLogin": ISODate("2025-12-06T12:00:00Z"),
"createdAt": ISODate("2025-10-15T10:30:00Z"),
"updatedAt": ISODate("2025-12-06T12:00:00Z"),
"preferences": {
"notificationChannels": ["PUSH", "EMAIL"],
"language": "it",
"theme": "LIGHT"
}
}

2.2 COLLEZIONE: `meters`

Descrizione: Contatori/sensori idrici associati agli utenti

Schema:
{
"_id": ObjectId,
"meterId": String, // ID univoco sensore (es. SN-001-BOLOGNA-001)
"userId": ObjectId, // Riferimento user (proprietario)
"meterType": String, // "DOMESTIC" | "COMMERCIAL" | "PUBLIC"
"location": String, // Ubicazione precisa (abitazione, azienda, ecc.)
"installationDate": ISODate, // Data installazione sensore
"deviceType": String, // Modello sensore (es. "ElectronicCounter-X100")
"serialNumber": String, // Numero seriale dispositivo

"status": String, // "ACTIVE" | "INACTIVE" | "MAINTENANCE"
"lastMaintenance": ISODate, // Ultima manutenzione
"communicationMethod": String, // "WIRELESS" | "WIRED" | "MANUAL"
"coordinates": {
"type": "Point",
"coordinates": [Number, Number] // [longitude, latitude] GeoJSON
},
"municipality": String, // Comune di appartenenza
"zone": String, // Zona geografica
"createdAt": ISODate,
"updatedAt": ISODate
}

Indici:
db.meters.createIndex({ "meterId": 1 }, { unique: true })
db.meters.createIndex({ "userId": 1 })
db.meters.createIndex({ "status": 1 })
db.meters.createIndex({ "municipality": 1 })
db.meters.createIndex({ "zone": 1 })
db.meters.createIndex({ "coordinates": "2dsphere" })

Esempio documento:
{
"_id": ObjectId("507f1f77bcf86cd799439012"),
"meterId": "SN-001-BO",
"userId": ObjectId("507f1f77bcf86cd799439011"),
"meterType": "DOMESTIC",
"location": "Via Roma 15, 40126 Bologna",
"installationDate": ISODate("2024-06-01T00:00:00Z"),
"deviceType": "ElectronicCounter-X100",
"serialNumber": "KAW53636844",
"status": "ACTIVE",
"lastMaintenance": ISODate("2025-09-15T10:00:00Z"),
"communicationMethod": "WIRELESS",
"municipality": "Bologna",
"zone": "Zone A - Centro",
"coordinates": {
"type": "Point",
"coordinates": [11.3426, 44.4949]
},
"createdAt": ISODate("2024-06-01T14:30:00Z"),
"updatedAt": ISODate("2025-12-06T12:00:00Z")
}

2.3 COLLEZIONE: `weekly_readings`

Descrizione: Letture consumi **settimanali** (da sensori)

Schema:
{
"_id": ObjectId,

"readingId": String, // ID univoco lettura (es. WR-20251206-001)
"meterId": ObjectId, // Riferimento meter
"userId": ObjectId, // Riferimento user (denormalizzazione)
"weekStartDate": ISODate, // Inizio settimana (lunedì)
"weekEndDate": ISODate, // Fine settimana (domenica)
"readingDate": ISODate, // Data lettura effettuata
"previousReading": Number, // Lettura precedente (mc)
"currentReading": Number, // Lettura corrente (mc)
"volumeConsumed": Number, // Litri consumati nella settimana
"volumeM3": Number, // m³ consumati
"readingType": String, // "TELELETTURA" | "MANUALE"
"dataQuality": String, // "VALID" | "ESTIMATED" | "SUSPICIOUS"
"cost": Number, // Costo stimato settimana (€)
"costBreakdown": {
"baseRate": Number, // Tariffa base (€/m³)
"consumptionCost": Number, // Costo consumo (€)
"fixedCharge": Number, // Canone fisso settimanale
"taxes": Number // Imposte
},
"syncedAt": ISODate, // Sincronizzazione dal sensore
"createdAt": ISODate,
"updatedAt": ISODate
}

Indici:
db.weekly_readings.createIndex({ "meterId": 1, "weekEndDate": -1 })
db.weekly_readings.createIndex({ "userId": 1, "weekEndDate": -1 })
db.weekly_readings.createIndex({ "readingId": 1 }, { unique: true })
db.weekly_readings.createIndex({ "weekStartDate": -1, "weekEndDate": -1 })

Esempio documento:
{
"_id": ObjectId("507f1f77bcf86cd799439030"),
"readingId": "WR-20250609",
"meterId": ObjectId("507f1f77bcf86cd799439012"),
"userId": ObjectId("507f1f77bcf86cd799439011"),
"weekStartDate": ISODate("2025-06-09T00:00:00Z"),
"weekEndDate": ISODate("2025-06-15T23:59:59Z"),
"readingDate": ISODate("2025-06-16T10:00:00Z"),
"previousReading": 79,
"currentReading": 82,
"volumeConsumed": 3000,
"volumeM3": 3,
"readingType": "TELELETTURA",
"dataQuality": "VALID",
"cost": 18.50,
"costBreakdown": {
"baseRate": 0.6433,
"consumptionCost": 1.93,
"fixedCharge": 2.31,
"taxes": 0.26

},
"syncedAt": ISODate("2025-06-16T11:15:00Z"),
"createdAt": ISODate("2025-06-16T11:16:00Z"),
"updatedAt": ISODate("2025-06-16T11:16:00Z")
}

2.4 COLLEZIONE: `consumption_history`

Descrizione: Storico consolidato consumi per grafici (aggregazione settimanale/mensile/annuale) - Basato su bollette reali

Schema:
{
"_id": ObjectId,
"userId": ObjectId,
"meterId": ObjectId,
"period": String, // "WEEKLY" | "MONTHLY" | "YEARLY"
"periodDate": ISODate, // Data inizio periodo
"periodStartDate": ISODate, // Data inizio lettura (es. 2024-12-31)
"periodEndDate": ISODate, // Data fine lettura (es. 2025-06-29)
"year": Number, // Anno
"month": Number, // Mese (1-12)
"week": Number, // Numero settimana (1-52)

// DATI CONSUMI DETTAGLIATI
"consumptionData": {
"previousReading": Number, // Lettura precedente (mc)
"currentReading": Number, // Lettura attuale (mc)
"totalVolumeMc": Number, // m³ totali periodo
"totalVolumeL": Number, // Litri totali
"readingType": String, // "TELELETTURA" | "MANUALE"
"averageDailyL": Number, // Media giornaliera (L/giorno)
"minDailyL": Number, // Consumo minimo giorno
"maxDailyL": Number // Consumo massimo giorno
},

// COMPOSIZIONE DEL CONSUMO PER FASCE TARIFFARIE
"consumptionByTariff": [
{
"tariffName": String, // Es. "Tariffa agevolata" | "Tariffa base" | "Tariffa maggiorata"
"minMc": Number, // Range minimo (mc)
"maxMc": Number, // Range massimo (mc)
"consumedMc": Number, // Consumo in questa fascia (mc)
"ratePerMc": Number, // Tariffa €/mc
"costEuro": Number // Costo questa fascia
}
],

// DETTAGLIO COSTI COME DA BOLLETTA
"costBreakdown": {
// ACQUA

"waterFixed": Number, // Quota fissa acqua (€)
"waterFixedPerWeek": Number, // Quota fissa acqua settimanale
"waterVariableTotal": Number, // Quota variabile acqua totale
"waterVariableAgevolata": Number, // Quota variabile acqua - Tariffa agevolata
"waterVariableBase": Number, // Quota variabile acqua - Tariffa base
"waterVariableMaggiorata": Number, // Quota variabile acqua - Tariffa maggiorata

```
// FOGNATURA
"sewerageFixed": Number,            // Quota fissa fognatura (€)
"sewerageVariable": Number,         // Quota variabile fognatura (€)

// DEPURAZIONE
"treatmentFixed": Number,           // Quota fissa depurazione (€)
"treatmentVariable": Number,        // Quota variabile depurazione (€)

// TOTALI
"totalBeforeTax": Number,           // Imponibile (somma sopra)
"taxRate": Number,                  // Aliquota IVA (es. 0.10 = 10%)
"taxAmount": Number,                // Importo IVA
"totalAmount": Number,              // Totale lordo

// ALTRI DATI
"previousBalance": Number,          // Saldo precedente (€)
"discount": Number,                 // Sconto applicato (€)
"finalAmount": Number,              // TOTALE DA PAGARE
"paidAmount": Number,               // Importo pagato (null se non pagato)
"paymentDate": ISODate              // Data pagamento
```

},

// METADATI BOLLETTA
"billMetadata": {
"billId": String, // ID fattura collegata
"billNumber": String, // Numero fattura (es. "25020032257")
"issueDate": ISODate, // Data emissione
"dueDate": ISODate, // Data scadenza
"billStatus": String, // "ISSUED" | "PAID" | "OVERDUE"
"billType": String, // "DOMESTIC" | "COMMERCIAL" | "PUBLIC"
"municipality": String, // Comune (es. "CIVEZZANO")
"meterSerialNumber": String, // Numero seriale contatore (es. "KAW53636844")
"meterCode": String, // Codice matricola (es. "A831C756")
"pdc": String // Punto di consegna (es. "H1512001010974")
},

// CALCOLI STATISTICI PER GRAFICI
"statistics": {
"trend": String, // "UP" | "DOWN" | "STABLE"
"trendPercentage": Number, // Percentuale aumento/diminuzione rispetto periodo precedente
"averageHourlyL": Number, // Media oraria
"anomalyDetected": Boolean, // Consumo anomalo?
"weeklyReadings": [ObjectId], // Array ObjectId letture settimanali incluse
"readingCount": Number // Numero letture settimanali nel periodo
},

"createdAt": ISODate,
"updatedAt": ISODate
}

Indici:
db.consumption_history.createIndex({ "userId": 1, "period": 1, "periodDate": -1 })
db.consumption_history.createIndex({ "meterId": 1, "period": 1, "periodDate": -1 })
db.consumption_history.createIndex({ "year": 1, "month": 1 })
db.consumption_history.createIndex({ "billMetadata.billId": 1 })
db.consumption_history.createIndex({ "periodStartDate": -1, "periodEndDate": -1 })

Esempio documento:
{
"_id": ObjectId("507f1f77bcf86cd799439021"),
"userId": ObjectId("507f1f77bcf86cd799439011"),
"meterId": ObjectId("507f1f77bcf86cd799439012"),
"period": "MONTHLY",
"periodDate": ISODate("2025-06-01T00:00:00Z"),
"periodStartDate": ISODate("2024-12-31T00:00:00Z"),
"periodEndDate": ISODate("2025-06-29T23:59:59Z"),
"year": 2025,
"month": 6,

"consumptionData": {
"previousReading": 18,
"currentReading": 96,
"totalVolumeMc": 78,
"totalVolumeL": 78000,
"readingType": "TELELETTURA",
"averageDailyL": 428.57,
"minDailyL": 300,
"maxDailyL": 550
},

"consumptionByTariff": [
{
"tariffName": "Tariffa agevolata",
"minMc": 0,
"maxMc": 110,
"consumedMc": 55,
"ratePerMc": 0.1930,

```
"costEuro": 10.53
},
{
"tariffName": "Tariffa base",
"minMc": 111,
"maxMc": 160,
"consumedMc": 23,
"ratePerMc": 0.6433,
"costEuro": 15.09
}
],

"costBreakdown": {
"waterFixed": 29.61,
"waterFixedPerWeek": 6.81,
"waterVariableTotal": 25.62,
"waterVariableAgevolata": 10.53,
"waterVariableBase": 15.09,
"waterVariableMaggiorata": 0,
"sewerageFixed": 4.21,
"sewerageVariable": 13.72,
"treatmentFixed": 0,
"treatmentVariable": 66.30,
"totalBeforeTax": 139.45,
"taxRate": 0.10,
"taxAmount": 13.95,
"totalAmount": 153.40,
"previousBalance": 0,
"discount": 0,
"finalAmount": 153.40,
"paidAmount": 153.40,
"paymentDate": ISODate("2025-10-06T00:00:00Z")
},

"billMetadata": {
"billId": "BILL-2025-06-001",
"billNumber": "25020032257",
"issueDate": ISODate("2025-09-01T00:00:00Z"),
"dueDate": ISODate("2025-10-06T00:00:00Z"),
"billStatus": "PAID",
"billType": "DOMESTIC",
"municipality": "CIVEZZANO",
"meterSerialNumber": "KAW53636844",
"meterCode": "A831C756",
"pdc": "H1512001010974"
},

"statistics": {
"trend": "DOWN",
"trendPercentage": -5.2,
"averageHourlyL": 17.86,
```

"anomalyDetected": false,
"weeklyReadings": [
ObjectId("507f1f77bcf86cd799439030"),
ObjectId("507f1f77bcf86cd799439031"),
ObjectId("507f1f77bcf86cd799439032"),
ObjectId("507f1f77bcf86cd799439033")
],
"readingCount": 4
},

"createdAt": ISODate("2025-09-01T10:00:00Z"),
"updatedAt": ISODate("2025-10-06T15:30:00Z")
}

2.5 COLLEZIONE: `tariffs`

Descrizione: Tabella costi e tariffe generali

Schema:
{
"_id": ObjectId,
"tariffId": String, // ID univoco tariffa (es. TARIFF-DOM-2025)
"name": String, // Nome tariffa (es. "Tariffa Domestica 2025")
"description": String, // Descrizione dettagliata
"type": String, // "DOMESTIC" | "COMMERCIAL" | "PUBLIC"
"municipality": String, // Comune applicazione
"effectiveDate": ISODate, // Data inizio validità
"expiryDate": ISODate, // Data fine validità
"baseRate": Number, // Tariffa base (€/m³)
"fixedCharge": Number, // Canone fisso mensile (€)
"fixedChargeWeekly": Number, // Canone fisso settimanale (€)
"consumptionBrackets": [
{
"minM3": Number, // Consumo minimo (m³)
"maxM3": Number, // Consumo massimo (m³) - null se illimitato
"ratePerM3": Number, // Tariffa per m³ in questo fascia (€/m³)
"description": String // Es. "Fascia 0-5 m³: tariffa ridotta"
}
],
"taxRate": Number, // Aliquota IVA (es. 0.10 = 10%)
"seasonalAdjustment": {
"summer": Number, // Moltiplicatore estivo (es. 1.1 = +10%)
"winter": Number // Moltiplicatore invernale
},
"isActive": Boolean, // Tariffa attualmente in uso?
"notes": String,
"createdAt": ISODate,
"updatedAt": ISODate
}

Indici:
db.tariffs.createIndex({ "tariffId": 1 }, { unique: true })
db.tariffs.createIndex({ "type": 1, "municipality": 1, "effectiveDate": -1 })
db.tariffs.createIndex({ "isActive": 1 })
db.tariffs.createIndex({ "effectiveDate": -1 })

Esempio documento:
{
"_id": ObjectId("507f1f77bcf86cd799439020"),
"tariffId": "TARIFF-DOM-BOLOGNA-2025",
"name": "Tariffa Domestica Bologna 2025",
"type": "DOMESTIC",
"municipality": "Bologna",
"effectiveDate": ISODate("2025-01-01T00:00:00Z"),
"expiryDate": ISODate("2025-12-31T23:59:59Z"),
"baseRate": 0.12,
"fixedCharge": 10.0,
"fixedChargeWeekly": 2.31,
"consumptionBrackets": [
{
"minM3": 0,
"maxM3": 5,
"ratePerM3": 0.12,
"description": "Fascia base 0-5 m³"
},
{
"minM3": 5,
"maxM3": 10,
"ratePerM3": 0.15,
"description": "Fascia intermedia 5-10 m³"
},
{
"minM3": 10,
"maxM3": null,
"ratePerM3": 0.20,
"description": "Fascia eccedenza >10 m³"
}
],
"taxRate": 0.10,
"isActive": true,
"createdAt": ISODate("2024-12-01T10:00:00Z"),
"updatedAt": ISODate("2024-12-01T10:00:00Z")
}

2.6 COLLEZIONE: bills

Descrizione: Fatture generate da letture settimanali aggregate

Schema:
{
"_id": ObjectId,

"billId": String, // ID univoco fattura (es. BILL-2025-12-001)
"userId": ObjectId,
"meterId": ObjectId,
"billingPeriodStart": ISODate, // Inizio periodo (es. 1° mese)
"billingPeriodEnd": ISODate, // Fine periodo (es. ultimo giorno mese)
"billingCycleWeeks": [ObjectId], // Array readingId settimanali incluse
"billIssueDate": ISODate,
"billDueDate": ISODate,
"billStatus": String, // "DRAFT" | "ISSUED" | "PAID" | "OVERDUE"
"consumptionData": {
"totalVolumeL": Number, // Litri totali periodo
"totalVolumeM3": Number, // m³ totali
"averageWeeklyL": Number, // Media settimanale
"maxWeeklyL": Number, // Settimana max
"weekCount": Number // N settimane fatturate
},
"costBreakdown": {
"fixedChargeTotal": Number, // Canone fisso totale (€)
"consumptionCostTotal": Number, // Costo consumo (€)
"taxesTotal": Number, // Imposte totali (€)
"previousBalance": Number, // Saldo precedente
"totalBeforeTax": Number, // Totale prima imposte
"totalAmount": Number, // Totale lordo (€)
"discount": Number, // Sconto applicato
"finalAmount": Number // Importo finale (€)
},
"tariffApplied": ObjectId, // Riferimento tariff usato
"paymentMethod": String, // "BANK_TRANSFER" | "CREDIT_CARD" | "PENDING"
"paymentDate": ISODate,
"notes": String,
"createdAt": ISODate,
"updatedAt": ISODate
}

Indici:
db.bills.createIndex({ "userId": 1, "billingPeriodEnd": -1 })
db.bills.createIndex({ "billId": 1 }, { unique: true })
db.bills.createIndex({ "billStatus": 1 })
db.bills.createIndex({ "billingPeriodEnd": -1 })
db.bills.createIndex({ "meterId": 1, "billingPeriodEnd": -1 })

Esempio documento:
{
"_id": ObjectId("507f1f77bcf86cd799439016"),
"billId": "BILL-2025-06-001",
"userId": ObjectId("507f1f77bcf86cd799439011"),
"meterId": ObjectId("507f1f77bcf86cd799439012"),
"billingPeriodStart": ISODate("2025-01-01T00:00:00Z"),
"billingPeriodEnd": ISODate("2025-06-30T23:59:59Z"),
"billingCycleWeeks": [
ObjectId("507f1f77bcf86cd799439030"),

ObjectId("507f1f77bcf86cd799439031"),
ObjectId("507f1f77bcf86cd799439032"),
ObjectId("507f1f77bcf86cd799439033")
],
"billIssueDate": ISODate("2025-09-01T10:00:00Z"),
"billDueDate": ISODate("2025-10-06T23:59:59Z"),
"billStatus": "PAID",
"consumptionData": {
"totalVolumeL": 78000,
"totalVolumeM3": 78,
"averageWeeklyL": 19500,
"maxWeeklyL": 21000,
"weekCount": 4
},
"costBreakdown": {
"fixedChargeTotal": 29.61,
"consumptionCostTotal": 25.62,
"taxesTotal": 13.95,
"previousBalance": 0,
"totalBeforeTax": 139.45,
"totalAmount": 153.40,
"discount": 0,
"finalAmount": 153.40
},
"tariffApplied": ObjectId("507f1f77bcf86cd799439020"),
"paymentMethod": "BANK_TRANSFER",
"paymentDate": ISODate("2025-10-06T00:00:00Z"),
"createdAt": ISODate("2025-09-01T10:00:00Z"),
"updatedAt": ISODate("2025-10-06T15:30:00Z")
}

2.7 COLLEZIONE: territorial_statistics

Descrizione: Dati aggregati anonimizzati per amministratori

Schema:
{
"_id": ObjectId,
"statisticsId": String,
"municipality": String,
"zone": String,
"statisticsDate": ISODate, // Data snapshot
"userCount": Number, // N utenti (min 10 privacy)
"meterCount": Number, // N contatori attivi
"consumptionMetrics": {
"totalVolumeL": Number,
"averagePerUserL": Number,
"averagePerWeekL": Number,
"medianConsumptionL": Number,
"stdDeviation": Number,
"minWeeklyL": Number,

"maxWeeklyL": Number
},
"costMetrics": {
"totalCost": Number,
"averagePerUser": Number,
"medianCost": Number
},
"meterTypeBreakdown": {
"domestic": Number,
"commercial": Number,
"public": Number
},
"heatmapIntensity": String, // "LOW" | "MEDIUM" | "HIGH" | "CRITICAL"
"userCountValid": Boolean, // >= 10 utenti per validità
"createdAt": ISODate,
"updatedAt": ISODate
}

Indici:
db.territorial_statistics.createIndex({ "municipality": 1, "statisticsDate": -1 })
db.territorial_statistics.createIndex({ "zone": 1, "statisticsDate": -1 })
db.territorial_statistics.createIndex({ "heatmapIntensity": 1 })

Esempio documento:
{
"_id": ObjectId("507f1f77bcf86cd799439019"),
"statisticsId": "STAT-BOLOGNA-ZONA-A-20251206",
"municipality": "Bologna",
"zone": "Zone A - Centro",
"statisticsDate": ISODate("2025-12-06T00:00:00Z"),
"userCount": 1245,
"meterCount": 1320,
"consumptionMetrics": {
"totalVolumeL": 315000,
"averagePerUserL": 253.0,
"averagePerWeekL": 10500.0,
"medianConsumptionL": 240.0,
"stdDeviation": 85.5,
"minWeeklyL": 45.0,
"maxWeeklyL": 890.0
},
"costMetrics": {
"totalCost": 4725.0,
"averagePerUser": 3.79,
"medianCost": 3.60
},
"meterTypeBreakdown": {
"domestic": 1100,
"commercial": 180,
"public": 40
},

"heatmapIntensity": "MEDIUM",
"userCountValid": true,
"createdAt": ISODate("2025-12-06T01:00:00Z"),
"updatedAt": ISODate("2025-12-06T01:00:00Z")
}

---

3. Relazioni tra Collezioni

users (1) ──→ (N) meters
users (1) ──→ (N) weekly_readings
users (1) ──→ (N) consumption_history
users (1) ──→ (N) bills

meters (1) ──→ (N) weekly_readings
meters (1) ──→ (N) consumption_history
meters (1) ──→ (N) bills

weekly_readings ──→ tariffs (lookup per calcoli costi)

consumption_history ──→ bills (one-to-one: same data, different purpose)
consumption_history ──→ weekly_readings (array riferimenti)

bills ──→ tariffs (quale tariffa usata)
bills ──→ weekly_readings (quali letture incluse)

territorial_statistics (aggregazione) → dati pubblici zones/municipalities

---

4. Flusso Dati Settimanale

1. Lettura Sensore (settimanale)
Sensore IoT → API Backend → weekly_readings collection
Salvataggio automatico ogni lunedì (o su schedule)

2. Calcolo Storico Consumi
weekly_readings → aggregazione per periodo → consumption_history
(Settimanale, mensile, annuale)

3. Generazione Bolletta Mensile
weekly_readings (4 settimane) + tariff lookup → bills collection
Ricalcolo costi automatico usando tariffe attuali

4. Grafici Dashboard
consumption_history (già aggregato) → visualizzazione istantanea
Query veloce senza calcoli real-time

---

5. Query Comuni per Dashboard

Storico Consumi Ultimi 12 Mesi (Mensile)

```
db.consumption_history.find({
userId: ObjectId("..."),
period: "MONTHLY"
}).sort({ periodDate: -1 }).limit(12)
```

Consumo Totale Annuale

```
db.consumption_history.aggregate([
{
$match: {
userId: ObjectId("..."),
period: "MONTHLY",
year: 2025
}
},
{
$group: {
_id: null,
totalConsumptionL: { $sum: "$consumptionData.totalVolumeL" },
totalSpent: { $sum: "$costBreakdown.finalAmount" },
avgMonthly: { $avg: "$costBreakdown.finalAmount" }
}
}
])
```

Confronto Fasce Tariffarie

```
db.consumption_history.aggregate([
{
$match: {
userId: ObjectId("..."),
period: "MONTHLY"
}
},
{
$unwind: "$consumptionByTariff"
},
{
consumptionByTariff.tariffName",
totalConsumed: { $sum: "$consumptionByTariff.consumedMc" },
totalCost: { $sum: "$consumptionByTariff.costEuro" },
avgRate: { $avg: "$consumptionByTariff.ratePerMc" }
}
}
])
```

Fatture Non Pagate

```
db.consumption_history.find({
userId: ObjectId("..."),
"billMetadata.billStatus": { $in: ["ISSUED", "OVERDUE"] },
"costBreakdown.paidAmount": null
}).sort({ "billMetadata.dueDate": 1 })
```

---

6. Sharding Strategy (Scalabilità Futura)

```
// Per collection voluminose - distribuzione per userId
db.weekly_readings.createIndex({ "userId": 1 })
// Shard key: { userId: 1 }

db.consumption_history.createIndex({ "userId": 1, "period": 1 })
// Shard key: { userId: 1 }
```

---

7. Validazione Schema (JSON Schema)

```
db.createCollection("weekly_readings", {
validator: {
$jsonSchema: {
bsonType: "object",
required: ["readingId", "meterId", "userId", "weekStartDate", "weekEndDate"],
properties: {
_id: { bsonType: "objectId" },
readingId: { bsonType: "string" },
meterId: { bsonType: "objectId" },
userId: { bsonType: "objectId" },
weekStartDate: { bsonType: "date" },
weekEndDate: { bsonType: "date" },
volumeConsumed: { bsonType: "double", minimum: 0 },
cost: { bsonType: "double", minimum: 0 },
readingType: { enum: ["TELELETTURA", "MANUALE"] }
}
}
}
})
```

---

8. Workflow Completo: Registrazione → Letture → Bolletta → Storico

1 Utente registrato

```
users.insertOne({
email: "mario.rossi@email.com",
userType: "PRIVATE",
firstName: "Mario"
```

...
})

2️⃣ Sensore associato

```
meters.insertOne({
meterId: "SN-001-BO",
userId: ObjectId("..."),
meterType: "DOMESTIC"
...
})
```

3️⃣ Letture settimanali (ogni lunedì)

```
weekly_readings.insertOne({
readingId: "WR-20250609",
meterId: ObjectId("..."),
userId: ObjectId("..."),
weekStartDate: ISODate("2025-06-09"),
weekEndDate: ISODate("2025-06-15"),
currentReading: 82,
volumeConsumed: 3000
})
```

4️⃣ Aggregazione fine mese

```
consumption_history.insertOne({
userId: ObjectId("..."),
period: "MONTHLY",
periodDate: ISODate("2025-06-01"),
consumptionData: { totalVolumeMc: 78 },
costBreakdown: { finalAmount: 153.40 }
})
```

5️⃣ Fattura mensile

```
bills.insertOne({
billId: "BILL-2025-06-001",
userId: ObjectId("..."),
billingCycleWeeks: [ObjectId(...), ObjectId(...), ...],
costBreakdown: { finalAmount: 153.40 }
})
```

---

9. Performance Ottimizzazioni

- **Indici compositi** per query frequenti (userId + date)
- **Denormalizzazione** su consumption_history per analytics veloci
- **Pre-aggregazione** territorial_statistics (batch job)
- **TTL opzionale** su bills (archiviazione dopo 7 anni)
- **Connection pooling** per 100K sensori

---

10. Conformità GDPR

- **Anonimizzazione**: territorial_statistics senza PII
- **Retention**: Bills 7 anni (obblighi fiscali)
- **Diritto all'oblio**: Soft-delete su users (status: "DELETED")
- **Crittografia**: AES-256 at-rest (MongoDB Enterprise)

---

11. Sommario Collezioni

Collezione	Documenti	Frequenza	Scopo
users	~2000	Statico	Gestione utenti
meters	~2500	Statico	Sensori IoT
weekly_readings	~520.000/anno	Settimanale	Letture grezze
consumption_history	~240.000	Mensile/Annuale	Analisi grafici
tariffs	~50	Annuale	Costi e tariffe
bills	~120.000	Mensile	Fatture
territorial_statistics	~5000	Giornaliero	Dati aggregati

Versione: 2.1
Data aggiornamento: 2025-12-06
MongoDB versione minima: 5.0
Frequenza letture: Settimanale
Basato su: Bollette reali AMAMBIENTE TN