

DOCUMENTACIÓN TÉCNICA

Tabla: jobnimbus_payments

Base de Datos:	JobNimbus Stamford
Sistema:	PostgreSQL 14+
Categoría:	Financiero - Pagos
Registros:	Transacciones de pago
Generado:	13/11/2025 22:05

1. RESUMEN EJECUTIVO

La tabla **jobnimbus_payments** es una entidad fundamental del sistema financiero que registra todas las transacciones de pago recibidas de clientes. Esta tabla actúa como el registro maestro de ingresos y permite el seguimiento completo del flujo de efectivo.

1.1 Propósito de la Tabla

- **Registro de Transacciones:** Captura cada pago recibido con monto, fecha y método
- **Conciliación de Invoices:** Vincula pagos con invoices para tracking de cuentas por cobrar
- **Métodos de Pago:** Documenta el método utilizado (efectivo, cheque, transferencia, tarjeta)
- **Referencias Bancarias:** Almacena números de cheque y referencias de transacciones
- **Auditoría Financiera:** Proporciona trail completo para reconciliación contable
- **Análisis de Flujo:** Permite análisis de patrones de pago y proyecciones de efectivo

1.2 Características Clave

Característica	Descripción
Total de Campos	20 campos de datos + 2 metadata
Campos JSONB	2 (related, owners)
Índices	5 índices de rendimiento
Constraints	UNIQUE(instance, jnid)
Precisión Decimal	DECIMAL(12,2) para montos
Validación	NOT NULL en campos críticos

2. ESTRUCTURA DE LA TABLA

2.1 Definición SQL

```
CREATE TABLE IF NOT EXISTS jobnimbus_payments ( id UUID PRIMARY KEY DEFAULT
uuid_generate_v4(), instance VARCHAR(20) NOT NULL, -- Identificación jnid VARCHAR(100) NOT
NULL, recid BIGINT, -- Fechas date_created BIGINT NOT NULL, date_updated BIGINT,
date_payment BIGINT, -- Monto amount DECIMAL(12,2) NOT NULL, -- Método de pago
payment_method VARCHAR(100), payment_type VARCHAR(100), -- Referencias check_number
VARCHAR(50), reference VARCHAR(255), memo TEXT, -- Relaciones con invoices invoice_jnid
VARCHAR(100), invoice_number VARCHAR(50), related JSONB, -- Ownership owners JSONB, --
Status is_active BOOLEAN DEFAULT TRUE, -- Metadata created_at TIMESTAMP DEFAULT
CURRENT_TIMESTAMP, updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP, UNIQUE(instance, jnid)
);
```

2.2 Resumen de Campos por Categoría

Categoría	Campos	Descripción
Identificación	4	UUID, instance, jnid, recid
Fechas	3	created, updated, payment
Montos	1	amount (principal)
Método de Pago	2	payment_method, payment_type
Referencias	3	check_number, reference, memo
Relaciones	3	invoice_jnid, invoice_number, related
Ownership	1	owners array
Status	1	is_active
Metadata	2	created_at, updated_at

3. CATÁLOGO DE CAMPOS

3.1 Campos de Identificación

Campo	Tipo	Descripción	Constraints
id	UUID	Identificador único interno generado automáticamente	PRIMARY KEY
instance	VARCHAR(20)	Instancia multi-tenant (stamford/guilford)	NOT NULL, INDEXED
jnid	VARCHAR(100)	JobNimbus ID único del pago	NOT NULL, UNIQUE
recid	BIGINT	Record ID numérico de JobNimbus	

3.2 Campos de Fechas

Campo	Tipo	Descripción	Constraints
date_created	BIGINT	Timestamp Unix de creación del registro	NOT NULL
date_updated	BIGINT	Timestamp Unix de última modificación	
date_payment	BIGINT	Timestamp Unix de fecha efectiva del pago	INDEXED

3.3 Montos

Campo	Tipo	Descripción	Constraints
amount	DECIMAL(12,2)	Monto total del pago recibido	NOT NULL

3.4 Método de Pago

Campo	Tipo	Descripción	Constraints
payment_method	VARCHAR(100)	Método de pago utilizado	INDEXED
payment_type	VARCHAR(100)	Tipo/categoría de pago	

3.5 Referencias

Campo	Tipo	Descripción	Constraints
check_number	VARCHAR(50)	Número de cheque si aplica	
reference	VARCHAR(255)	Referencia de transacción bancaria	
memo	TEXT	Notas o memo del pago	

3.6 Relaciones con Invoices

Campo	Tipo	Descripción	Constraints
<code>invoice_jnid</code>	VARCHAR(100)	JNID del invoice principal asociado	INDEXED
<code>invoice_number</code>	VARCHAR(50)	Número del invoice asociado	
<code>related</code>	JSONB	Array de todas las entidades relacionadas	

3.7 Ownership

Campo	Tipo	Descripción	Constraints
<code>owners</code>	JSONB	Array de usuarios propietarios del pago	

3.8 Status

Campo	Tipo	Descripción	Constraints
<code>is_active</code>	BOOLEAN	Indica si el pago está activo	DEFAULT TRUE

3.9 Metadata del Sistema

Campo	Tipo	Descripción	Constraints
<code>created_at</code>	TIMESTAMP	Timestamp de creación en PostgreSQL	DEFAULT NOW()
<code>updated_at</code>	TIMESTAMP	Timestamp de última actualización	DEFAULT NOW()

4. ÍNDICES Y OPTIMIZACIÓN

4.1 Índices Definidos

Nombre	Tipo	Columna(s)	Propósito
idx_jn_payments_instance	B-Tree	instance	Filtrado multi-tenant
idx_jn_payments_jnid	B-Tree	jnid	Búsqueda rápida por ID JobNimbus
idx_jn_payments_date_payment	B-Tree	date_payment	Reportes por período
idx_jn_payments_invoice	B-Tree	invoice_jnid	JOIN con invoices
idx_jn_payments_method	B-Tree	payment_method	Análisis por método de pago

4.2 Constraints y Validaciones

- **PRIMARY KEY:** id (UUID único generado automáticamente)
- **UNIQUE:** (instance, jnid) - Previene duplicados por instancia
- **NOT NULL:** instance, jnid, date_created, amount - Campos obligatorios
- **DEFAULT:** is_active = TRUE, created_at/updated_at = CURRENT_TIMESTAMP
- **DECIMAL(12,2):** Precisión de 2 decimales para montos financieros

4.3 Recomendaciones de Performance

- Usar **date_payment** indexado para queries de reportes por período
- Filtrar siempre por **instance** para aprovechar particionamiento lógico
- Usar **invoice_jnid** para JOINS rápidos con tabla de invoices
- Indexar **payment_method** permite análisis eficiente por tipo de pago
- El constraint UNIQUE previene pagos duplicados por error de sistema

5. CAMPOS JSONB - ESTRUCTURA Y USO

5.1 Campo: related (Relaciones)

El campo **related** almacena un array de objetos JSON que representan todas las entidades relacionadas con el pago. Típicamente incluye el invoice, contact y job asociados.

Ejemplo de Estructura:

```
[ { "id": "mea5f606nznul7hekvlw6", "type": "invoice", "name": "#3106", "number": "3106",  
  "email": null, "subject": null }, { "id": "ltwy0ds3xgrblq7rdlt1hf6", "type": "contact",  
  "name": "Jacqueline Almanzar", "number": "1144", "email": null, "subject": null }, { "id":  
  "ltq9a7bscyrxamk5ltdnaan", "type": "job", "name": "36 west ave roof replacement",  
  "number": "1146", "email": null, "subject": null } ]
```

Queries útiles para campo related:

```
-- Pagos relacionados con un invoice específico SELECT * FROM jobnimbus_payments WHERE  
related @> '["type": "invoice", "id": "mea5f606nznul7hekvlw6"]'; -- Pagos relacionados  
con un job SELECT * FROM jobnimbus_payments WHERE related @> '["type": "job"]'; --  
Extraer nombre del contact relacionado SELECT jnid, amount, related -> 1 ->> 'name' as  
contact_name FROM jobnimbus_payments WHERE related -> 1 ->> 'type' = 'contact';
```

5.2 Campo: owners (Propietarios)

El campo **owners** almacena un array de objetos que identifican a los usuarios propietarios del pago (típicamente sales reps o managers).

Ejemplo de Estructura:

```
[ { "id": "melnhiah2xkslc7r8uya48"}, { "id": "ltonct898ai3n4cgz9tels6"} ]
```

Queries útiles para campo owners:

```
-- Pagos de un owner específico SELECT * FROM jobnimbus_payments WHERE owners @> '["id":  
"ltonct898ai3n4cgz9tels6"]'; -- Contar owners por pago SELECT jnid, amount,  
jsonb_array_length(owners) as owner_count FROM jobnimbus_payments; -- Pagos con múltiples  
owners SELECT * FROM jobnimbus_payments WHERE jsonb_array_length(owners) > 1;
```

6. EJEMPLOS DE QUERIES

6.1 Reportes Financieros

Total de pagos por mes

```
SELECT DATE_TRUNC('month', TO_TIMESTAMP(date_payment)) as mes, COUNT(*) as cantidad_pagos, SUM(amount) as total_recaudado FROM jobnimbus_payments WHERE instance = 'stamford' AND is_active = TRUE AND date_payment > EXTRACT(EPOCH FROM NOW() - INTERVAL '6 months') GROUP BY DATE_TRUNC('month', TO_TIMESTAMP(date_payment)) ORDER BY mes DESC;
```

Top 10 pagos más grandes

```
SELECT jnid, amount, TO_TIMESTAMP(date_payment) as fecha_pago, payment_method, invoice_number FROM jobnimbus_payments WHERE instance = 'stamford' AND is_active = TRUE ORDER BY amount DESC LIMIT 10;
```

6.2 Análisis por Método de Pago

Distribución por método de pago

```
SELECT payment_method, COUNT(*) as cantidad, SUM(amount) as total, ROUND(AVG(amount), 2) as promedio FROM jobnimbus_payments WHERE instance = 'stamford' AND is_active = TRUE AND date_payment > EXTRACT(EPOCH FROM NOW() - INTERVAL '3 months') GROUP BY payment_method ORDER BY total DESC;
```

6.3 Conciliación con Invoices

Pagos por invoice con balance

```
SELECT i.number as invoice_no, i.total as invoice_total, COALESCE(SUM(p.amount), 0) as total_pagado, i.total - COALESCE(SUM(p.amount), 0) as balance_pendiente FROM jobnimbus_invoices i LEFT JOIN jobnimbus_payments p ON p.invoice_jnid = i.jnid AND p.is_active = TRUE WHERE i.instance = 'stamford' AND i.is_active = TRUE GROUP BY i.jnid, i.number, i.total HAVING i.total - COALESCE(SUM(p.amount), 0) > 0 ORDER BY balance_pendiente DESC;
```

6.4 Análisis de Flujo de Efectivo

Proyección de flujo semanal

```
SELECT DATE_TRUNC('week', TO_TIMESTAMP(date_payment)) as semana, COUNT(*) as num_transacciones, SUM(amount) as ingreso_total, ROUND(AVG(amount), 2) as ticket_promedio FROM jobnimbus_payments WHERE instance = 'stamford' AND is_active = TRUE AND date_payment > EXTRACT(EPOCH FROM NOW() - INTERVAL '12 weeks') GROUP BY DATE_TRUNC('week', TO_TIMESTAMP(date_payment)) ORDER BY semana DESC;
```


7. APÉNDICE

7.1 Métodos de Pago Comunes

method_id	Método	Descripción
0	Cash	Efectivo
1	Credit Card	Tarjeta de crédito
2	Debit Card	Tarjeta de débito
3	Check	Cheque
4	Wire Transfer	Transferencia bancaria
5	ACH	Transferencia electrónica

7.2 Buenas Prácticas

- **Validación de Montos:** Siempre validar que amount > 0 antes de insertar
- **Referencias Únicas:** Usar reference para almacenar IDs de transacciones bancarias
- **Auditoría:** Nunca eliminar pagos, usar is_active = FALSE
- **Conciliación:** Verificar que la suma de pagos no exceda el total del invoice
- **Fechas:** date_payment puede diferir de date_created (fecha de aplicación vs recepción)
- **Multi-tenant:** Siempre filtrar por instance en queries de producción

7.3 Consideraciones de Seguridad

- Los pagos contienen información financiera sensible - aplicar controles de acceso
- Registrar todos los cambios en tabla de auditoría para compliance
- Validar permisos antes de permitir modificaciones o eliminaciones
- Encriptar información sensible como números de tarjeta (si se almacenan)
- Implementar reconciliación automática con sistema bancario

7.4 Campos de Timestamp

Los campos de fecha utilizan timestamps Unix (segundos desde epoch). Para convertir a fechas legibles en PostgreSQL, usar la función **TO_TIMESTAMP()**.

```
-- Convertir timestamp a fecha SELECT jnid, TO_TIMESTAMP(date_payment) as fecha_pago,  
TO_CHAR(TO_TIMESTAMP(date_payment), 'YYYY-MM-DD HH24:MI:SS') as fecha_formateada FROM  
jobnimbus_payments LIMIT 5;
```