

---

# DOCUMENTACIÓN TÉCNICA

## TABLA CONTACTS

### Base de Datos JobNimbus Stamford

<b>Tabla:</b>	jobnimbus_contacts
<b>Instancia:</b>	Stamford
<b>Total de Campos:</b>	45+
<b>Tipo:</b>	Tabla Transaccional
<b>Fecha:</b>	13/11/2025

---

# 1. RESUMEN EJECUTIVO

La tabla **jobnimbus\_contacts** es una tabla fundamental en la base de datos de JobNimbus Stamford que almacena toda la información de contactos (clientes, prospectos, subcontratistas, etc.). Esta tabla contiene **45+ campos** que cubren información personal, datos de contacto, direcciones, relaciones y campos personalizados flexibles.

Métrica	Valor	Descripción
Total de Campos	45+	Campos estructurados y JSONB
Campos JSONB	3	related, tags, custom_fields
Índices	8	Para optimización de consultas
Campos de Teléfono	5	Primary, Home, Mobile, Work, Fax
Direcciones	2 conjuntos	Principal y Billing

---

## 2. ESTRUCTURA DE LA TABLA

A continuación se presenta la definición SQL completa de la tabla `jobnimbus_contacts`, incluyendo todos los campos, tipos de datos y constraints.

```
CREATE TABLE IF NOT EXISTS jobnimbus_contacts (  
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),  
  instance VARCHAR(20) NOT NULL,  
  
  -- Identificación  
  jnid VARCHAR(100) NOT NULL,  
  recid BIGINT,  
  display_name VARCHAR(255),  
  
  -- Información personal  
  first_name VARCHAR(100),  
  last_name VARCHAR(100),  
  company VARCHAR(255),  
  email VARCHAR(255),  
  
  -- Teléfonos  
  phone_primary VARCHAR(50),  
  phone_home VARCHAR(50),  
  phone_mobile VARCHAR(50),  
  phone_work VARCHAR(50),  
  phone_fax VARCHAR(50),  
  
  -- Dirección principal  
  address_line1 VARCHAR(255),  
  address_line2 VARCHAR(255),  
  city VARCHAR(100),  
  state_text VARCHAR(100),  
  state_code VARCHAR(10),  
  zip VARCHAR(20),  
  country VARCHAR(100),  
  
  -- Dirección de billing  
  billing_line1 VARCHAR(255),  
  billing_line2 VARCHAR(255),  
  billing_city VARCHAR(100),  
  billing_state VARCHAR(100),  
  billing_zip VARCHAR(20),  
  
  -- Metadata  
  date_created BIGINT NOT NULL,  
  date_updated BIGINT,  
  record_type_name VARCHAR(100),  
  record_type BIGINT,  
  
  -- Relaciones y datos flexibles  
  related JSONB,  
  tags JSONB,
```

---

```
custom_fields JSONB,  
  
-- Status  
is_active BOOLEAN DEFAULT TRUE,  
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  
UNIQUE(instance, jnid)  
);
```

### 3. CATÁLOGO COMPLETO DE CAMPOS

#### 3.1 Campos de Identificación

Campos únicos que identifican cada contacto en el sistema

Campo	Tipo	Descripción	Constraints
id	UUID	Identificador único interno de PostgreSQL	PRIMARY KEY
instance	VARCHAR(20)	Identificador de instancia multi-tenant (stamford/guilford)	NOT NULL
jnid	VARCHAR(100)	JobNimbus ID único del contacto	NOT NULL, UNIQUE
recid	BIGINT	Record ID numérico de JobNimbus	
display_name	VARCHAR(255)	Nombre completo para mostrar	

#### 3.2 Información Personal

Datos personales y de empresa del contacto

Campo	Tipo	Descripción	Constraints
first_name	VARCHAR(100)	Nombre de pila	
last_name	VARCHAR(100)	Apellido	
company	VARCHAR(255)	Nombre de la empresa	
email	VARCHAR(255)	Dirección de correo electrónico	INDEXED

#### 3.3 Números de Teléfono

Múltiples números de contacto telefónico

Campo	Tipo	Descripción	Constraints
phone_primary	VARCHAR(50)	Número de teléfono principal	
phone_home	VARCHAR(50)	Teléfono de casa	
phone_mobile	VARCHAR(50)	Teléfono móvil/celular	
phone_work	VARCHAR(50)	Teléfono del trabajo	
phone_fax	VARCHAR(50)	Número de fax	

### 3.4 Dirección Principal

Dirección física principal del contacto

Campo	Tipo	Descripción	Constraints
address_line1	VARCHAR(255)	Línea 1 de dirección (calle y número)	
address_line2	VARCHAR(255)	Línea 2 de dirección (apartamento, suite)	
city	VARCHAR(100)	Ciudad	INDEXED
state_text	VARCHAR(100)	Nombre completo del estado	
state_code	VARCHAR(10)	Código de estado (ej: CT, NY)	
zip	VARCHAR(20)	Código postal	
country	VARCHAR(100)	País	

### 3.5 Dirección de Facturación (Billing)

Dirección alternativa para facturación si difiere de la principal

Campo	Tipo	Descripción	Constraints
billing_line1	VARCHAR(255)	Línea 1 de dirección de billing	
billing_line2	VARCHAR(255)	Línea 2 de dirección de billing	
billing_city	VARCHAR(100)	Ciudad de billing	
billing_state	VARCHAR(100)	Estado de billing	
billing_zip	VARCHAR(20)	Código postal de billing	

### 3.6 Metadata y Fechas

Información de control y auditoría

Campo	Tipo	Descripción	Constraints
date_created	BIGINT	Fecha de creación (Unix timestamp)	NOT NULL, INDEXED
date_updated	BIGINT	Fecha de última actualización (Unix timestamp)	
record_type_name	VARCHAR(100)	Nombre del tipo de registro (Customer, Lead, etc.)	

---

record_type	BIGINT	ID numérico del tipo de registro	
-------------	--------	----------------------------------	--

### 3.7 Campos JSONB Flexibles

Campos de datos estructurados pero flexibles

Campo	Tipo	Descripción	Constraints
related	JSONB	Array de relaciones con jobs, estimates, etc.	
tags	JSONB	Array de etiquetas asociadas al contacto	
custom_fields	JSONB	Campos personalizados definidos por el usuario	

### 3.8 Campos de Estado y Control

Indicadores de estado y control del registro

Campo	Tipo	Descripción	Constraints
is_active	BOOLEAN	Indica si el contacto está activo	DEFAULT TRUE
created_at	TIMESTAMP	Timestamp de creación en PostgreSQL	DEFAULT CURRENT_TIMESTAMP
updated_at	TIMESTAMP	Timestamp de última actualización en PostgreSQL	DEFAULT CURRENT_TIMESTAMP

---

## 4. ÍNDICES Y OPTIMIZACIÓN

Los índices son estructuras de datos que mejoran la velocidad de las operaciones de consulta en una tabla de base de datos. La tabla `jobnimbus_contacts` cuenta con **8 índices** estratégicamente diseñados para optimizar las consultas más frecuentes.

Índice	Campo(s)	Tipo	Propósito
<code>idx_jn_contacts_instance</code>	<code>instance</code>	B-Tree	Filtrado por instancia (stamford/guilford)
<code>idx_jn_contacts_jnid</code>	<code>jnid</code>	B-Tree	Búsqueda rápida por JobNimbus ID
<code>idx_jn_contacts_email</code>	<code>email</code>	B-Tree	Búsqueda por email de contacto
<code>idx_jn_contacts_name</code>	<code>last_name, first_name</code>	B-Tree	Búsqueda por nombre completo
<code>idx_jn_contacts_company</code>	<code>company</code>	B-Tree	Búsqueda por empresa
<code>idx_jn_contacts_date_created</code>	<code>date_created</code>	B-Tree	Ordenamiento y filtrado por fecha

### 4.1 Comandos SQL de Creación de Índices

```
CREATE INDEX idx_jn_contacts_instance ON jobnimbus_contacts(instance);
CREATE INDEX idx_jn_contacts_jnid ON jobnimbus_contacts(jnid);
CREATE INDEX idx_jn_contacts_email ON jobnimbus_contacts(email);
CREATE INDEX idx_jn_contacts_name ON jobnimbus_contacts(last_name, first_name);
CREATE INDEX idx_jn_contacts_company ON jobnimbus_contacts(company);
CREATE INDEX idx_jn_contacts_date_created ON jobnimbus_contacts(date_created);
```



---

## 5. CAMPOS JSONB FLEXIBLES

Los campos JSONB permiten almacenar datos estructurados pero flexibles en formato JSON. Estos campos son ideales para almacenar arrays de objetos y datos que pueden variar entre registros sin necesidad de modificar el esquema de la tabla.

### 5.1 Campo 'related' - Relaciones del Contacto

El campo **related** almacena un array de objetos que representan las relaciones del contacto con otros registros en JobNimbus (jobs, estimates, invoices, etc.).

```
{
  "related": [
    {
      "id": "job_jnid_123",
      "type": "job",
      "number": "1820",
      "display_name": "Roof Replacement - 123 Main St"
    },
    {
      "id": "estimate_jnid_456",
      "type": "estimate",
      "number": "EST-2025-001"
    }
  ]
}
```

### 5.2 Campo 'tags' - Etiquetas

El campo **tags** almacena un array de etiquetas asociadas al contacto para categorización y búsqueda rápida.

```
{
  "tags": [
    {
      "id": "tag_123",
      "name": "VIP Customer",
      "color": "#FF5733"
    },
    {
      "id": "tag_456",
      "name": "Referral Source"
    }
  ]
}
```

---

## 5.3 Campo 'custom\_fields' - Campos Personalizados

El campo **custom\_fields** permite almacenar campos personalizados definidos por el usuario que no están en el esquema estándar de la tabla.

```
{
  "custom_fields": {
    "preferred_contact_time": "Morning",
    "referral_source": "Google Ads",
    "customer_since": "2023-05-15",
    "preferred_language": "English",
    "special_notes": "Prefers email communication"
  }
}
```

## 5.4 Ejemplos de Consultas JSONB

```
-- Buscar contactos con una etiqueta específica
SELECT jnid, display_name
FROM jobnimbus_contacts
WHERE tags @> ' [{"name": "VIP Customer"}]';

-- Buscar contactos relacionados con un job específico
SELECT jnid, display_name
FROM jobnimbus_contacts
WHERE related @> ' [{"type": "job", "id": "job_jnid_123"}]';

-- Extraer un campo personalizado
SELECT jnid, display_name,
       custom_fields->>'preferred_language' as language
FROM jobnimbus_contacts
WHERE custom_fields->>'preferred_language' = 'Spanish';
```

---

## 6. EJEMPLOS DE DATOS REALES

A continuación se presentan ejemplos de registros reales de la tabla `jobnimbus_contacts` para ilustrar cómo se almacenan los datos en la práctica.

### 6.1 Ejemplo: Contacto Completo con Dirección

```
{
  "jnid": "mhxl5scuqxs4f42p4mpl68w",
  "instance": "stamford",
  "number": "2038",
  "display_name": "Gregory Zaccheo",
  "first_name": "Gregory",
  "last_name": "Zaccheo",
  "email": "gzaccheo752@gmail.com",
  "phone_mobile": "914-555-0123",
  "address_line1": "70 Oakland Avenue",
  "city": "Harrison",
  "state_code": "NY",
  "zip": "10528",
  "country": "United States",
  "record_type_name": "Customer",
  "status_name": "New",
  "date_created": 1763047835,
  "date_updated": 1763047922,
  "is_active": true
}
```

### 6.2 Ejemplo: Contacto Empresarial

```
{
  "jnid": "mhxgygl4zbeox3xcwpjlawg",
  "instance": "stamford",
  "number": "2037",
  "display_name": "Jonathan McLeod",
  "first_name": "Jonathan",
  "last_name": "McLeod",
  "company": "McLeod Properties LLC",
  "email": "jonathan.e.mcleod@gmail.com",
  "phone_work": "203-555-0199",
  "phone_mobile": "203-555-0198",
  "address_line1": "485 Smith Avenue",
  "city": "New Haven",
  "state_code": "CT",
  "zip": "06511",
  "record_type_name": "Customer",
}
```

---

```
"status_name": "Active",  
"date_created": 1763040774,  
"is_active": true  
}
```

---

## 7. RELACIONES CON OTRAS TABLAS

La tabla `jobnimbus_contacts` tiene relaciones con múltiples tablas en la base de datos. Estas relaciones se establecen principalmente a través del campo JSONB `'related'` y mediante foreign keys en otras tablas.

Tabla Relacionada	Campo de Unión	Tipo de Relación	Descripción
<code>jobnimbus_jobs</code>	<code>related JSONB</code>	1:N	Un contacto puede tener múltiples jobs
<code>jobnimbus_estimates</code>	<code>related JSONB</code>	1:N	Un contacto puede tener múltiples estimates
<code>jobnimbus_invoices</code>	<code>related JSONB</code>	1:N	Un contacto puede tener múltiples invoices
<code>jobnimbus_activities</code>	<code>related JSONB</code>	1:N	Un contacto puede tener múltiples activities
<code>jobnimbus_tasks</code>	<code>related JSONB</code>	1:N	Un contacto puede tener múltiples tasks

### 7.1 Ejemplo de JOIN con Jobs

```
SELECT
  c.display_name as contact_name,
  c.email,
  c.phone_mobile,
  j.number as job_number,
  j.display_name as job_name,
  j.status_name as job_status
FROM jobnimbus_contacts c
CROSS JOIN LATERAL jsonb_array_elements(c.related) as rel
JOIN jobnimbus_jobs j ON j.jnid = rel->>'id'
WHERE c.instance = 'stamford'
      AND rel->>'type' = 'job'
      AND c.is_active = TRUE
ORDER BY c.display_name;
```

---

## 8. EJEMPLOS DE QUERIES COMUNES

### 8.1 Listar Contactos Activos de Stamford

Obtener todos los contactos activos ordenados por nombre

```
SELECT
    number,
    display_name,
    email,
    phone_mobile,
    city,
    state_code,
    record_type_name
FROM jobnimbus_contacts
WHERE instance = 'stamford'
    AND is_active = TRUE
ORDER BY last_name, first_name;
```

### 8.2 Buscar Contacto por Email

Búsqueda rápida de contacto por dirección de email

```
SELECT
    jnid,
    display_name,
    email,
    phone_mobile,
    address_line1,
    city,
    state_code
FROM jobnimbus_contacts
WHERE email = 'gzaccheo752@gmail.com'
    AND instance = 'stamford';
```

### 8.3 Contactos Creados en un Rango de Fechas

Listar contactos creados en un período específico

```
SELECT
    number,
    display_name,
    email,
    TO_TIMESTAMP(date_created) as created_date,
    record_type_name
```

```
FROM jobnimbus_contacts
WHERE instance = 'stamford'
    AND date_created >= EXTRACT(EPOCH FROM '2025-11-01'::timestamp)
    AND date_created <= EXTRACT(EPOCH FROM '2025-11-30'::timestamp)
ORDER BY date_created DESC;
```

## 8.4 Contactos por Ciudad

Agrupar y contar contactos por ciudad

```
SELECT
    city,
    state_code,
    COUNT(*) as contact_count
FROM jobnimbus_contacts
WHERE instance = 'stamford'
    AND is_active = TRUE
    AND city IS NOT NULL
GROUP BY city, state_code
ORDER BY contact_count DESC, city;
```

## 8.5 Contactos con Múltiples Teléfonos

Buscar contactos que tienen más de un número telefónico

```
SELECT
    display_name,
    phone_primary,
    phone_mobile,
    phone_home,
    phone_work
FROM jobnimbus_contacts
WHERE instance = 'stamford'
    AND (
        (phone_primary IS NOT NULL AND phone_primary != '') +
        (phone_mobile IS NOT NULL AND phone_mobile != '') +
        (phone_home IS NOT NULL AND phone_home != '') +
        (phone_work IS NOT NULL AND phone_work != '')
    ) > 1;
```

## 8.6 Contactos sin Email

Identificar contactos que no tienen email registrado

```
SELECT
    number,
```

---

```
    display_name,  
    phone_mobile,  
    city,  
    record_type_name  
FROM jobnimbus_contacts  
WHERE instance = 'stamford'  
    AND (email IS NULL OR email = '')  
    AND is_active = TRUE  
ORDER BY date_created DESC;
```

## 8.7 Actualizar Información de Contacto

Ejemplo de actualización de datos de un contacto

```
UPDATE jobnimbus_contacts  
SET  
    phone_mobile = '203-555-9999',  
    email = 'newemail@example.com',  
    updated_at = CURRENT_TIMESTAMP  
WHERE jnid = 'mhl5scuqxs4f42p4mp168w'  
    AND instance = 'stamford';
```

## 8.8 Contactos por Tipo de Registro

Agrupar contactos por su tipo (Customer, Lead, etc.)

```
SELECT  
    record_type_name,  
    COUNT(*) as total,  
    COUNT(CASE WHEN is_active THEN 1 END) as active,  
    COUNT(CASE WHEN NOT is_active THEN 1 END) as inactive  
FROM jobnimbus_contacts  
WHERE instance = 'stamford'  
GROUP BY record_type_name  
ORDER BY total DESC;
```



---

## 9. MEJORES PRÁCTICAS

### 9.1 Validación de Datos

- Validar formato de email antes de insertar
- Normalizar números telefónicos a un formato consistente
- Validar códigos de estado y códigos postales
- Verificar que instance sea siempre "stamford" o "guilford"

### 9.2 Uso de Índices

- Usar el índice idx\_jn\_contacts\_email para búsquedas por email
- Usar idx\_jn\_contacts\_name para búsquedas por nombre
- Siempre filtrar por instance en consultas multi-tenant
- Aprovechar idx\_jn\_contacts\_date\_created para consultas por rango de fechas

### 9.3 Manejo de Campos JSONB

- Usar operadores JSONB (@>, ->, ->>) para consultas eficientes
- Evitar consultas que requieran escanear todo el JSONB
- Considerar crear índices GIN para consultas JSONB frecuentes
- Validar estructura JSON antes de insertar en campos JSONB

### 9.4 Actualización de Registros

- Siempre actualizar updated\_at al modificar un registro
- Usar transacciones para actualizaciones que afecten múltiples tablas
- No eliminar registros físicamente, usar is\_active = FALSE
- Mantener auditoría de cambios importantes

---

## 10. APÉNDICE

### 10.1 Glosario de Términos

Término	Definición
JNID	JobNimbus Unique Identifier - Identificador único de JobNimbus
RECID	Record ID - ID numérico secuencial en JobNimbus
Instance	Identificador de instancia multi-tenant (stamford/guilford)
JSONB	JSON Binary - Formato binario de PostgreSQL para almacenar JSON
Unix Timestamp	Número de segundos desde el 1 de enero de 1970
B-Tree Index	Índice de árbol balanceado para búsquedas eficientes
GIN Index	Generalized Inverted Index para datos JSONB

### 10.2 Conversión de Unix Timestamps

Los campos de fecha en JobNimbus se almacenan como Unix timestamps (segundos desde 1970-01-01). Para convertirlos a fechas legibles en PostgreSQL:

```
-- Convertir Unix timestamp a fecha
SELECT
  jnid,
  display_name,
  TO_TIMESTAMP(date_created) as created_date,
  TO_CHAR(TO_TIMESTAMP(date_created), 'YYYY-MM-DD HH24:MI:SS') as formatted_date
FROM jobnimbus_contacts
WHERE instance = 'stamford'
LIMIT 5;

-- Convertir fecha a Unix timestamp
SELECT EXTRACT(EPOCH FROM '2025-11-13 10:30:00'::timestamp) as unix_timestamp;
```