

DOCUMENTACIÓN TÉCNICA

Tabla: jobnimbus_tasks

Base de Datos:	JobNimbus Stamford
Sistema:	PostgreSQL 14+
Categoría:	Gestión de Tareas
Registros:	Tareas y actividades programadas
Generado:	13/11/2025 22:13

1. RESUMEN EJECUTIVO

La tabla **jobnimbus_tasks** es el sistema central de gestión de tareas y actividades programadas. Permite la asignación, seguimiento y control de tareas relacionadas con jobs, contacts, estimates y otras entidades del sistema.

1.1 Propósito de la Tabla

- **Gestión de Tareas:** Asignación y seguimiento de tareas a usuarios
- **Programación:** Calendarización de llamadas, reuniones y seguimientos
- **Priorización:** Sistema de prioridades para organizar el trabajo
- **Tracking de Tiempo:** Estimación y registro de horas trabajadas
- **Productividad:** Medición de completación y eficiencia del equipo
- **Workflow:** Control de flujo de trabajo y seguimiento de pendientes

1.2 Características Clave

Característica	Descripción
Total de Campos	22 campos de datos + 2 metadata
Campos JSONB	2 (related, owners)
Índices	6 índices de rendimiento
Constraints	UNIQUE(instance, jnid)
Prioridades	0-5 (0=baja, 5=urgente)
Time Tracking	estimated_hours, actual_hours

2. ESTRUCTURA DE LA TABLA

2.1 Definición SQL

```
CREATE TABLE IF NOT EXISTS jobnimbus_tasks ( id UUID PRIMARY KEY DEFAULT
uuid_generate_v4(), instance VARCHAR(20) NOT NULL, -- Identificación jnid VARCHAR(100) NOT
NULL, recid BIGINT, -- Task info title VARCHAR(255), description TEXT, -- Fechas
date_created BIGINT NOT NULL, date_updated BIGINT, date_start BIGINT, date_end BIGINT,
date_due BIGINT, -- Prioridad y estado priority INT DEFAULT 0, status VARCHAR(100),
completed BOOLEAN DEFAULT FALSE, -- Asignación assigned_to_jnid VARCHAR(100),
assigned_to_name VARCHAR(255), owners JSONB, -- Relaciones related JSONB, -- Tiempo
estimado y real estimated_hours DECIMAL(6,2), actual_hours DECIMAL(6,2), -- Metadata
is_active BOOLEAN DEFAULT TRUE, 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
Información	2	title, description
Fechas	5	created, updated, start, end, due
Prioridad/Estado	3	priority, status, completed
Asignación	3	assigned_to_jnid, assigned_to_name, owners
Relaciones	1	related (JSONB)
Time Tracking	2	estimated_hours, actual_hours
Metadata	3	is_active, 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 de la tarea	NOT NULL, UNIQUE, INDEXED
recid	BIGINT	Record ID numérico de JobNimbus	

3.2 Información de la Tarea

Campo	Tipo	Descripción	Constraints
title	VARCHAR(255)	Título o nombre de la tarea	
description	TEXT	Descripción detallada de la tarea	

3.3 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_start	BIGINT	Timestamp Unix de inicio programado	
date_end	BIGINT	Timestamp Unix de fin programado	
date_due	BIGINT	Timestamp Unix de fecha límite	INDEXED

3.4 Prioridad y Estado

Campo	Tipo	Descripción	Constraints
priority	INT	Nivel de prioridad (0-5, 0=baja, 5=urgente)	DEFAULT 0, INDEXED
status	VARCHAR(100)	Estado de la tarea	INDEXED
completed	BOOLEAN	Indica si la tarea está completada	DEFAULT FALSE

3.5 Asignación

Campo	Tipo	Descripción	Constraints
assigned_to_jnid	VARCHAR(100)	JNID del usuario asignado principal	INDEXED

assigned_to_name	VARCHAR(255)	Nombre del usuario asignado	
owners	JSONB	Array de usuarios propietarios/responsables	

3.6 Relaciones

Campo	Tipo	Descripción	Constraints
related	JSONB	Array de entidades relacionadas (jobs, contacts, etc.)	

3.7 Time Tracking

Campo	Tipo	Descripción	Constraints
estimated_hours	DECIMAL(6,2)	Horas estimadas para completar	
actual_hours	DECIMAL(6,2)	Horas reales trabajadas	

3.8 Metadata del Sistema

Campo	Tipo	Descripción	Constraints
is_active	BOOLEAN	Indica si el registro está activo	DEFAULT TRUE
created_at	TIMESTAMP	Timestamp de creación en PostgreSQL	DEFAULT NOW()
updated_at	TIMESTAMP	Timestamp de última actualización	DEFAULT NOW()

4. ÍNDICES Y OPTIMIZACIÓN

4.1 Índices Definidos

Nombre	Tipo	Columna(s)	Propósito
idx_jn_tasks_instance	B-Tree	instance	Filtrado multi-tenant
idx_jn_tasks_jnid	B-Tree	jnid	Búsqueda rápida por ID
idx_jn_tasks_assigned_to	B-Tree	assigned_to_jnid	Tareas por usuario asignado
idx_jn_tasks_status	B-Tree	status	Filtrado por estado
idx_jn_tasks_priority	B-Tree	priority	Ordenamiento por prioridad
idx_jn_tasks_date_due	B-Tree	date_due	Tareas por vencer

4.2 Sistema de Prioridades

Valor	Nivel	Descripción
0	Baja	Tareas de baja prioridad, sin urgencia
1	Normal-Baja	Prioridad ligeramente elevada
2	Normal	Prioridad estándar
3	Normal-Alta	Requiere atención pronto
4	Alta	Requiere atención prioritaria
5	Urgente	Máxima prioridad, atención inmediata

5. CAMPOS JSONB - ESTRUCTURA Y USO

5.1 Campo: related (Entidades Relacionadas)

El campo **related** almacena un array de objetos JSON que representan las entidades relacionadas con la tarea. Típicamente incluye jobs, contacts, estimates o invoices.

Ejemplo de Estructura:

```
[ { "id": "mh3es0bg8mp8tj64vix9o5s", "type": "job", "name": "19 Cleveland Street",  
  "number": "1896", "email": null, "subject": null }, { "id": "mh3er9rtlru0tbs5w5feh1",  
  "type": "contact", "name": "Dione", "number": "1936", "email": null, "subject": null } ]
```

Queries útiles para campo related:

```
-- Tareas relacionadas con un job específico SELECT * FROM jobnimbus_tasks WHERE related @>  
'[{"type": "job", "id": "mh3es0bg8mp8tj64vix9o5s"}]'; -- Tareas asociadas a contacts  
SELECT * FROM jobnimbus_tasks WHERE related @> '[{"type": "contact"}]'; -- Extraer nombre  
del job relacionado SELECT title, related -> 0 ->> 'name' as job_name, related -> 0 ->>  
'number' as job_number FROM jobnimbus_tasks WHERE related -> 0 ->> 'type' = 'job';
```

5.2 Campo: owners (Propietarios)

El campo **owners** almacena un array de objetos que identifican a los usuarios propietarios o responsables de la tarea.

Ejemplo de Estructura:

```
[ {"id": "mg5ajt9klpl1t1libn9lsk2g"}, {"id": "ltn4m8iyu23g9vpm52b0m67"} ]
```

Queries útiles para campo owners:

```
-- Tareas de un owner específico SELECT * FROM jobnimbus_tasks WHERE owners @> '[{"id":  
"mg5ajt9klpl1t1libn9lsk2g"}]'; -- Tareas con múltiples owners SELECT * FROM jobnimbus_tasks  
WHERE jsonb_array_length(owners) > 1; -- Contar tareas por usuario SELECT owners -> 0 ->>  
'id' as user_id, COUNT(*) as task_count FROM jobnimbus_tasks WHERE owners IS NOT NULL GROUP  
BY owners -> 0 ->> 'id';
```

6. EJEMPLOS DE QUERIES

6.1 Gestión de Tareas Pendientes

Tareas pendientes por usuario

```
SELECT title, priority, TO_TIMESTAMP(date_due) as fecha_limite, assigned_to_name FROM
jobnimbus_tasks WHERE instance = 'stamford' AND completed = FALSE AND is_active = TRUE AND
assigned_to_jnid = 'user_jnid_here' ORDER BY priority DESC, date_due ASC;
```

Tareas vencidas

```
SELECT title, assigned_to_name, TO_TIMESTAMP(date_due) as fecha_limite, EXTRACT(EPOCH FROM
NOW()) - date_due as segundos_vencido FROM jobnimbus_tasks WHERE instance = 'stamford' AND
completed = FALSE AND is_active = TRUE AND date_due < EXTRACT(EPOCH FROM NOW()) ORDER BY
date_due ASC;
```

6.2 Análisis de Productividad

Tasa de completación por usuario

```
SELECT assigned_to_name, COUNT(*) as total_tareas, SUM(CASE WHEN completed THEN 1 ELSE 0
END) as completadas, ROUND(100.0 * SUM(CASE WHEN completed THEN 1 ELSE 0 END) / COUNT(*),
1) as porcentaje_completacion FROM jobnimbus_tasks WHERE instance = 'stamford' AND
is_active = TRUE AND date_created > EXTRACT(EPOCH FROM NOW() - INTERVAL '30 days') GROUP BY
assigned_to_name ORDER BY porcentaje_completacion DESC;
```

6.3 Time Tracking

Comparación estimado vs real

```
SELECT title, estimated_hours, actual_hours, actual_hours - estimated_hours as diferencia,
CASE WHEN estimated_hours > 0 THEN ROUND(100.0 * (actual_hours - estimated_hours) /
estimated_hours, 1) ELSE NULL END as porcentaje_diferencia FROM jobnimbus_tasks WHERE
instance = 'stamford' AND completed = TRUE AND estimated_hours IS NOT NULL AND actual_hours
IS NOT NULL ORDER BY ABS(actual_hours - estimated_hours) DESC LIMIT 20;
```

6.4 Análisis por Prioridad

Distribución de tareas por prioridad

```
SELECT priority, CASE priority WHEN 0 THEN 'Baja' WHEN 1 THEN 'Normal-Baja' WHEN 2 THEN
'Normal' WHEN 3 THEN 'Normal-Alta' WHEN 4 THEN 'Alta' WHEN 5 THEN 'Urgente' END as nivel,
COUNT(*) as cantidad, SUM(CASE WHEN completed THEN 1 ELSE 0 END) as completadas FROM
jobnimbus_tasks WHERE instance = 'stamford' AND is_active = TRUE GROUP BY priority ORDER BY
priority DESC;
```


7. APÉNDICE

7.1 Tipos de Tareas Comunes

record_type	Nombre	Descripción
2	Phone Call	Llamada telefónica
3	Meeting	Reunión o cita
4	Follow-up	Seguimiento de cliente
5	To-Do	Tarea general
6	Site Visit	Visita al sitio

7.2 Buenas Prácticas

- **Priorización:** Asignar prioridades consistentemente para organizar trabajo
- **Fechas Límite:** Siempre establecer date_due para tareas críticas
- **Time Tracking:** Registrar estimated_hours y actual_hours para análisis
- **Asignación Clara:** Siempre asignar a un usuario específico (assigned_to_jnid)
- **Completación:** Marcar completed = TRUE al finalizar tareas
- **Relaciones:** Vincular tareas a jobs/contacts para contexto completo

7.3 Workflow de Gestión de Tareas

Creación: Usuario crea tarea con título, descripción, prioridad y fecha límite

Asignación: Tarea se asigna a usuario específico (assigned_to_jnid)

Programación: Se establecen date_start y date_end para calendarización

Ejecución: Usuario trabaja en la tarea, registra actual_hours

Completación: Se marca completed = TRUE al finalizar

Análisis: Se compara estimated_hours vs actual_hours para mejora continua

7.4 Campos de Timestamp

```
-- Tareas de la próxima semana SELECT title, priority, TO_TIMESTAMP(date_due) as
fecha_limite FROM jobnimbus_tasks WHERE instance = 'stamford' AND completed = FALSE AND
date_due BETWEEN EXTRACT(EPOCH FROM NOW()) AND EXTRACT(EPOCH FROM NOW() + INTERVAL '7
days') ORDER BY priority DESC, date_due ASC;
```