

DOCUMENTACIÓN TÉCNICA

Tabla: jobnimbus_activities

Base de Datos:	JobNimbus Stamford
Sistema:	PostgreSQL 14+
Categoría:	Auditoría - Activity Trail
Registros:	Historial de actividades
Generado:	13/11/2025 22:09

1. RESUMEN EJECUTIVO

La tabla **jobnimbus_activities** es el registro maestro de auditoría que captura todas las acciones, cambios y eventos que ocurren en el sistema JobNimbus. Funciona como un audit trail completo proporcionando trazabilidad total de todas las operaciones.

1.1 Propósito de la Tabla

- **Audit Trail:** Registro completo de todas las acciones realizadas en el sistema
- **Cambios de Status:** Tracking detallado de cambios de estado en jobs, estimates e invoices
- **Comunicaciones:** Historial de emails, llamadas y mensajes enviados
- **Modificaciones:** Log de actualizaciones y cambios en entidades
- **Trazabilidad:** Quién hizo qué, cuándo y en qué registro
- **Compliance:** Cumplimiento de requisitos de auditoría y regulación

1.2 Características Clave

Característica	Descripción
Total de Campos	19 campos de datos + 2 metadata
Campos JSONB	1 (related entities)
Índices	6 índices de rendimiento
Constraints	UNIQUE(instance, jnid)
Tipos de Actividad	27+ tipos diferentes (Email, Status Changed, etc.)
Volumen	Alto - miles de registros por mes

2. ESTRUCTURA DE LA TABLA

2.1 Definición SQL

```
CREATE TABLE IF NOT EXISTS jobnimbus_activities ( id UUID PRIMARY KEY DEFAULT
uuid_generate_v4(), instance VARCHAR(20) NOT NULL, -- Identificación jnid VARCHAR(100) NOT
NULL, recid BIGINT, -- Tipo de actividad activity_type VARCHAR(100), activity_name
VARCHAR(255), -- Fechas date_created BIGINT NOT NULL, date_updated BIGINT, date_start
BIGINT, date_end BIGINT, -- Descripción description TEXT, notes TEXT, -- Usuario que
realizó la actividad created_by_jnid VARCHAR(100), created_by_name VARCHAR(255), --
Relaciones related JSONB, related_to_jnid VARCHAR(100), related_to_type VARCHAR(100), --
Status status VARCHAR(100), completed BOOLEAN DEFAULT FALSE, -- 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
Tipo	2	activity_type, activity_name
Fechas	4	created, updated, start, end
Descripción	2	description, notes
Usuario	2	created_by_jnid, created_by_name
Relaciones	3	related (JSONB), related_to_jnid, related_to_type
Status	2	status, completed
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 actividad	NOT NULL, UNIQUE, INDEXED
recid	BIGINT	Record ID numérico de JobNimbus	

3.2 Tipo de Actividad

Campo	Tipo	Descripción	Constraints
activity_type	VARCHAR(100)	Tipo de actividad (Email, Status Changed, etc.)	INDEXED
activity_name	VARCHAR(255)	Nombre descriptivo de la actividad	

3.3 Campos de Fechas

Campo	Tipo	Descripción	Constraints
date_created	BIGINT	Timestamp Unix de creación del registro	NOT NULL, INDEXED
date_updated	BIGINT	Timestamp Unix de última modificación	
date_start	BIGINT	Timestamp Unix de inicio de actividad programada	
date_end	BIGINT	Timestamp Unix de fin de actividad	

3.4 Descripción y Notas

Campo	Tipo	Descripción	Constraints
description	TEXT	Descripción detallada de la actividad	
notes	TEXT	Notas adicionales o detalles de cambios	

3.5 Usuario Creador

Campo	Tipo	Descripción	Constraints
created_by_jnid	VARCHAR(100)	JNID del usuario que creó la actividad	INDEXED
created_by_name	VARCHAR(255)	Nombre del usuario creador	

3.6 Relaciones

Campo	Tipo	Descripción	Constraints
related	JSONB	Array de entidades relacionadas (jobs, contacts, etc.)	
related_to_jnid	VARCHAR(100)	JNID de entidad principal relacionada	INDEXED
related_to_type	VARCHAR(100)	Tipo de entidad relacionada (job, contact, etc.)	

3.7 Status

Campo	Tipo	Descripción	Constraints
status	VARCHAR(100)	Estado de la actividad	
completed	BOOLEAN	Indica si la actividad está completada	DEFAULT FALSE

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_activities_instance	B-Tree	instance	Filtrado multi-tenant
idx_jn_activities_jnid	B-Tree	jnid	Búsqueda rápida por ID
idx_jn_activities_type	B-Tree	activity_type	Filtrado por tipo de actividad
idx_jn_activities_date_created	B-Tree	date_created	Queries por período
idx_jn_activities_created_by	B-Tree	created_by_jnid	Actividades por usuario
idx_jn_activities_related_to	B-Tree	related_to_jnid	JOIN con entidades

4.2 Consideraciones de Performance

- Tabla de **alto volumen** - miles de registros por mes requiere particionamiento
- Usar **date_created** indexado para queries por período de tiempo
- Filtrar por **activity_type** para análisis de tipos específicos
- Index en **created_by_jnid** permite reportes eficientes por usuario
- Index en **related_to_jnid** optimiza timeline de jobs/contacts
- Considerar particionamiento por mes para mejorar rendimiento

5. TIPOS DE ACTIVIDADES

5.1 Tipos Comunes de Actividades

record_type	Nombre	Descripción
3	Email	Email enviado o recibido
19	Status Changed	Cambio de estado en job/estimate/invoice
1	Note	Nota agregada al registro
2	Call	Llamada telefónica registrada
4	Meeting	Reunión o cita agendada
5	Task	Tarea creada o completada
6	Document	Documento adjuntado
7	Photo	Foto agregada
8	SMS	Mensaje de texto

5.2 Ejemplos de Actividades por Tipo

Activity Type: Email (record_type = 3) Note: "Subject: G.A Castro Construction has sent you a Estimate/Proposal Sent To: customer@email.com" Activity Type: Status Changed (record_type = 19) Note: "Job Updated Status: Appointment Scheduled => Pending Customer Approval" Activity Type: Note (record_type = 1) Note: "Customer requested color change from gray to black"

6. CAMPO JSONB - ESTRUCTURA Y USO

6.1 Campo: related (Entidades Relacionadas)

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

Ejemplo de Estructura:

```
[ { "id": "mhnjtcc67sxmgiaw9gswcta", "type": "job", "name": "422 High Ridge Road",  
  "number": "1921", "email": null, "subject": null } ]
```

Queries útiles para campo related:

```
-- Actividades relacionadas con un job específico SELECT * FROM jobnimbus_activities WHERE  
related @> '["type": "job", "id": "mhnjtcc67sxmgiaw9gswcta"]'; -- Timeline completo de  
un job SELECT date_created, activity_type, created_by_name, notes FROM  
jobnimbus_activities WHERE related_to_jnid = 'mhnjtcc67sxmgiaw9gswcta' ORDER BY  
date_created DESC; -- Actividades de cambios de status SELECT TO_TIMESTAMP(date_created)  
as fecha, related -> 0 ->> 'name' as entity_name, notes FROM jobnimbus_activities WHERE  
activity_type = 'Status Changed' AND instance = 'stamford' ORDER BY date_created DESC  
LIMIT 10;
```


7. EJEMPLOS DE QUERIES

7.1 Timeline de Entidades

Timeline completo de un job

```
SELECT TO_TIMESTAMP(date_created) as fecha, activity_type, created_by_name as usuario,  
LEFT(notes, 100) as resumen FROM jobnimbus_activities WHERE related_to_jnid =  
'job_jnid_here' AND instance = 'stamford' AND is_active = TRUE ORDER BY date_created DESC;
```

7.2 Análisis de Actividades

Actividades por tipo en el último mes

```
SELECT activity_type, COUNT(*) as cantidad, COUNT(DISTINCT created_by_jnid) as  
usuarios_unicos FROM jobnimbus_activities WHERE instance = 'stamford' AND date_created >  
EXTRACT(EPOCH FROM NOW() - INTERVAL '30 days') GROUP BY activity_type ORDER BY cantidad  
DESC;
```

7.3 Productividad de Usuarios

Top usuarios por actividad

```
SELECT created_by_name, COUNT(*) as total_actividades, COUNT(DISTINCT  
DATE(TO_TIMESTAMP(date_created))) as dias_activos, ROUND(COUNT(*)::DECIMAL /  
COUNT(DISTINCT DATE(TO_TIMESTAMP(date_created))), 1) as actividades_por_dia FROM  
jobnimbus_activities WHERE instance = 'stamford' AND date_created > EXTRACT(EPOCH FROM  
NOW() - INTERVAL '30 days') AND created_by_name IS NOT NULL GROUP BY created_by_name ORDER  
BY total_actividades DESC LIMIT 10;
```

7.4 Análisis de Cambios de Status

Cambios de status más frecuentes

```
SELECT notes, COUNT(*) as frecuencia FROM jobnimbus_activities WHERE instance = 'stamford'  
AND activity_type = 'Status Changed' AND date_created > EXTRACT(EPOCH FROM NOW() -  
INTERVAL '90 days') GROUP BY notes ORDER BY frecuencia DESC LIMIT 20;
```

8. APÉNDICE

8.1 Buenas Prácticas

- **Retención de Datos:** Implementar política de archivado para actividades antiguas (>2 años)
- **Particionamiento:** Considerar particionar por mes para mejorar performance
- **Indexación:** Los 6 índices existentes son críticos para queries eficientes
- **Auditoría:** Nunca eliminar activities - son registro permanente
- **Timeline:** Usar related_to_jnid para construir timelines de entidades
- **Búsqueda:** Implementar full-text search en notes para búsqueda eficiente

8.2 Uso de Timestamps

Los campos de fecha utilizan timestamps Unix (segundos desde epoch). Para queries por período, convertir a timestamps PostgreSQL usando **TO_TIMESTAMP()**.

```
-- Actividades del último mes SELECT * FROM jobnimbus_activities WHERE date_created >
EXTRACT(EPOCH FROM NOW() - INTERVAL '30 days') AND instance = 'stamford' ORDER BY
date_created DESC; -- Convertir a fecha legible SELECT TO_TIMESTAMP(date_created) as
fecha, TO_CHAR(TO_TIMESTAMP(date_created), 'YYYY-MM-DD HH24:MI') as fecha_formateada,
activity_type, notes FROM jobnimbus_activities LIMIT 5;
```

8.3 Consideraciones de Volumen

- Tabla de **alto volumen** - puede crecer a millones de registros
- Implementar **particionamiento mensual** después de 1M registros
- Archivar actividades antiguas a tabla histórica
- Monitorear tamaño de índices y hacer REINDEX periódicamente
- Considerar índice parcial en is_active = TRUE para queries activas