**Nakshatra-Yogam Mappings: Database Design & Integration Documentation**

**Overview**

This document provides comprehensive documentation for the nakshatra\_yogam\_mappings reference table implementation in the astrological database system. This design enables efficient storage and retrieval of yogam values associated with different nakshatras on different days of the week.

**Table Design**

**Structure of nakshatra\_yogam\_mappings**

A dedicated reference table was created to store the relationship between nakshatras and their associated yogams for each day of the week:

CREATE TABLE IF NOT EXISTS nakshatra\_yogam\_mappings (

id SERIAL PRIMARY KEY,

nakshatra\_name TEXT,

nakshatra\_name\_english TEXT,

sunday\_yogam TEXT,

monday\_yogam TEXT,

tuesday\_yogam TEXT,

wednesday\_yogam TEXT,

thursday\_yogam TEXT,

friday\_yogam TEXT,

saturday\_yogam TEXT,

created\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()

);

**Key Features**

* **Bilingual Support**: Both Tamil (nakshatra\_name) and English (nakshatra\_name\_english) names are stored
* **Day-specific Yogams**: Each nakshatra has specific yogam values for each day of the week
* **Timestamps**: Creation time is tracked for audit purposes
* **Unique Identifier**: Each mapping has a unique sequential ID

**Data Population**

The table is populated with 27 nakshatras, each with its specific yogam for each day of the week:

INSERT INTO nakshatra\_yogam\_mappings (

nakshatra\_name,

nakshatra\_name\_english,

sunday\_yogam,

monday\_yogam,

tuesday\_yogam,

wednesday\_yogam,

thursday\_yogam,

friday\_yogam,

saturday\_yogam

) VALUES

('அசுவினி', 'Ashwini', 'சித்தயோகம்', 'சித்தயோகம்', 'சித்தயோகம்', 'மரணயோகம்', 'அமிர்தயோகம்', 'அமிர்தயோகம்', 'சித்தயோகம்'),

-- All other values as specified

('ரேவதி', 'Revathi', 'அமிர்தயோகம்', 'சித்தயோகம்', 'சித்தயோகம்', 'மரணயோகம்', 'சித்தயோகம்', 'அமிர்தயோகம்', 'மரணயோகம்');

**Database Optimization**

The following indexes are created to optimize query performance:

CREATE INDEX IF NOT EXISTS idx\_nakshatra\_yogam\_mappings\_name

ON nakshatra\_yogam\_mappings(nakshatra\_name);

CREATE INDEX IF NOT EXISTS idx\_nakshatra\_yogam\_mappings\_english\_name

ON nakshatra\_yogam\_mappings(nakshatra\_name\_english);

**Access Methods**

**Helper Function**

A dedicated function provides easy access to yogam values based on nakshatra and day:

CREATE OR REPLACE FUNCTION get\_nakshatra\_yogam(nakshatra\_name TEXT, day\_name TEXT)

RETURNS TEXT AS $$

DECLARE

yogam TEXT;

day\_column TEXT;

BEGIN

-- Convert day name to lowercase and remove any spaces or day suffix

day\_name := LOWER(TRIM(day\_name));

day\_name := REGEXP\_REPLACE(day\_name, '\s+', '');

-- Handle different formats of day names

IF day\_name IN ('sun', 'sunday', 'ஞாயிறு') THEN

day\_column := 'sunday\_yogam';

ELSIF day\_name IN ('mon', 'monday', 'திங்கள்') THEN

day\_column := 'monday\_yogam';

ELSIF day\_name IN ('tue', 'tuesday', 'செவ்வாய்') THEN

day\_column := 'tuesday\_yogam';

ELSIF day\_name IN ('wed', 'wednesday', 'புதன்') THEN

day\_column := 'wednesday\_yogam';

ELSIF day\_name IN ('thu', 'thursday', 'வியாழன்') THEN

day\_column := 'thursday\_yogam';

ELSIF day\_name IN ('fri', 'friday', 'வெள்ளி') THEN

day\_column := 'friday\_yogam';

ELSIF day\_name IN ('sat', 'saturday', 'சனி') THEN

day\_column := 'saturday\_yogam';

ELSE

RETURN NULL;

END IF;

EXECUTE format('

SELECT %I

FROM nakshatra\_yogam\_mappings

WHERE nakshatra\_name = $1 OR nakshatra\_name\_english = $1

LIMIT 1', day\_column)

INTO yogam

USING nakshatra\_name;

RETURN yogam;

END;

$$ LANGUAGE plpgsql;

**Database View (Optional)**

A view can be created for simplified access:

CREATE OR REPLACE VIEW panchangam\_with\_yogam AS

SELECT

dp.\*,

get\_nakshatra\_yogam(

dp.main\_nakshatra,

TO\_CHAR(dp.date, 'Day')

) AS nakshatra\_yogam

FROM

daily\_panchangam dp;

**Integration with Existing Systems**

**Method 1: Direct Table Queries**

Query the reference table directly:

-- Example: Get yogam for a specific nakshatra on a specific day

SELECT monday\_yogam

FROM nakshatra\_yogam\_mappings

WHERE nakshatra\_name = 'அசுவினி' OR nakshatra\_name\_english = 'Ashwini';

**Method 2: Function-Based Access (Recommended)**

Use the helper function for flexible access:

-- Get yogam for current date and nakshatra

SELECT

get\_nakshatra\_yogam('Ashwini', TO\_CHAR(CURRENT\_DATE, 'Day')) AS today\_yogam;

-- Get yogam for a specific date in daily\_panchangam

SELECT

date,

main\_nakshatra,

get\_nakshatra\_yogam(main\_nakshatra, TO\_CHAR(date, 'Day')) as nakshatra\_yogam

FROM daily\_panchangam

WHERE date = '2023-05-06';

**Method 3: JSONB Integration**

Update the existing JSONB field in your daily\_panchangam table:

-- Update the JSONB field with nakshatra\_yogam information

UPDATE daily\_panchangam dp

SET yoga = jsonb\_set(

COALESCE(dp.yoga, '{}'::jsonb),

'{nakshatra\_yogam}',

to\_jsonb(get\_nakshatra\_yogam(

dp.main\_nakshatra,

TO\_CHAR(dp.date, 'Day')

))

)

WHERE dp.date >= '2023-01-01';

**Application Integration**

**Backend Access (Node.js/JavaScript Example)**

// Fetch yogam using Supabase client

const getNakshatraYogam = async (nakshatra, date) => {

const { data, error } = await supabase

.rpc('get\_nakshatra\_yogam', {

nakshatra\_name: nakshatra,

day\_name: new Date(date).toLocaleDateString('en-US', { weekday: 'long' })

});

if (error) console.error('Error fetching yogam:', error);

return data;

};

// Alternative: Query the view directly

const getPanchangamWithYogam = async (date) => {

const { data, error } = await supabase

.from('panchangam\_with\_yogam')

.select('\*')

.eq('date', date);

if (error) console.error('Error fetching panchangam:', error);

return data;

};

**Frontend Display Example**

// React component example

function NakshatraYogamDisplay({ nakshatra, date }) {

const [yogam, setYogam] = useState(null);

useEffect(() => {

async function fetchYogam() {

const yogamData = await getNakshatraYogam(nakshatra, date);

setYogam(yogamData);

}

fetchYogam();

}, [nakshatra, date]);

return (

<div className="yogam-display">

<h3>Yogam for {nakshatra} on {new Date(date).toDateString()}</h3>

{yogam ? <p>{yogam}</p> : <p>Loading...</p>}

</div>

);

}

**Future Extensibility**

**Adding New Nakshatras**

INSERT INTO nakshatra\_yogam\_mappings (

nakshatra\_name,

nakshatra\_name\_english,

sunday\_yogam,

monday\_yogam,

tuesday\_yogam,

wednesday\_yogam,

thursday\_yogam,

friday\_yogam,

saturday\_yogam

) VALUES (

'நக்ஷத்திரம்', 'New Nakshatra',

'சித்தயோகம்', 'அமிர்தயோகம்', 'மரணயோகம்', 'சித்தயோகம்',

'அமிர்தயோகம்', 'சித்தயோகம்', 'மரணயோகம்'

);

**Updating Existing Mappings**

UPDATE nakshatra\_yogam\_mappings

SET monday\_yogam = 'அமிர்தயோகம்'

WHERE nakshatra\_name = 'அசுவினி' OR nakshatra\_name\_english = 'Ashwini';

**Maintenance and Best Practices**

1. **Regular Backups**: Include this table in regular database backup routines
2. **Version Control**: Track schema changes and data updates in version control
3. **Access Control**: Set appropriate permissions for this reference table
4. **Audit Trail**: Consider implementing audit triggers for tracking changes
5. **Documentation**: Keep this documentation updated with any schema changes

**Benefits of This Design**

1. **Normalized Data Structure**: Eliminates redundancy and improves data integrity
2. **Improved Performance**: Optimized for quick lookups with appropriate indexes
3. **Extensibility**: Easy to add or modify yogam mappings without affecting other tables
4. **Flexibility**: Multiple access methods support different use cases
5. **Maintainability**: Centralized management of nakshatra-yogam relationships
6. **Bilingual Support**: Works with both Tamil and English nakshatra names

**Troubleshooting**

**Common Issues and Solutions**

1. **Query returning NULL**:
   * Check that nakshatra name matches exactly (case-sensitive)
   * Verify day name format is recognized by the function
2. **Performance concerns**:
   * Ensure indexes are created and not fragmented
   * Consider caching frequently accessed values
3. **Missing yogam values**:
   * Verify nakshatra exists in the mapping table
   * Check for typos in nakshatra names

*This documentation was created on May 6, 2025.*

**Amruthathi Yogam System Documentation**

**Overview**

The Amruthathi Yogam (nakshatra yogam) system is an important astrological feature of the TamilJyotish Daily Panchangam application. It determines the quality or nature of a particular day for a person born under a specific nakshatra (birth star), providing guidance on auspicious and inauspicious times for activities.

**Database Schema**

**Table: nakshatra\_yogam\_mappings**

This reference table stores the relationship between nakshatras and their associated yogams for each day of the week.

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| id | serial | Primary key |
| nakshatra\_name | text | Nakshatra name in Tamil (e.g., ரோகிணி, அஸ்வினி) |
| nakshatra\_name\_english | text | Nakshatra name in English (e.g., Rohini, Ashwini) |
| sunday\_yogam | text | Yogam for Sunday (e.g., சித்தயோகம், அமிர்தயோகம், மரணயோகம்) |
| monday\_yogam | text | Yogam for Monday |
| tuesday\_yogam | text | Yogam for Tuesday |
| wednesday\_yogam | text | Yogam for Wednesday |
| thursday\_yogam | text | Yogam for Thursday |
| friday\_yogam | text | Yogam for Friday |
| saturday\_yogam | text | Yogam for Saturday |
| created\_at | timestamp with timezone | Record creation timestamp |

**Types of Yogams**

The three primary types of yogams used in the system are:

1. **சித்தயோகம் (Siddha Yogam)** - Favorable for accomplishment and fulfillment of desires
2. **அமிர்தயோகம் (Amrutha Yogam)** - Highly favorable, promising excellent results
3. **மரணயோகம் (Marana Yogam)** - Unfavorable, suggesting caution for important activities

**Database Function**

The application uses a Supabase RPC function to retrieve the appropriate yogam:

sql

CREATE OR REPLACE FUNCTION get\_nakshatra\_yogam(nakshatra\_name TEXT, day\_name TEXT)

RETURNS TEXT AS $$

DECLARE

yogam TEXT;

day\_column TEXT;

normalized\_name TEXT;

BEGIN

*-- Normalize nakshatra name to handle spelling variations*

IF nakshatra\_name = 'அஸ்தம்' THEN

normalized\_name := 'ஹஸ்தம்';

ELSE

normalized\_name := nakshatra\_name;

END IF;

*-- Convert day name to lowercase and handle different formats*

day\_name := LOWER(TRIM(day\_name));

*-- Determine which column to query based on day name*

IF day\_name IN ('sun', 'sunday', 'ஞாயிறு', 'ஞாயிற்றுக்கிழமை') THEN

day\_column := 'sunday\_yogam';

ELSIF day\_name IN ('mon', 'monday', 'திங்கள்', 'திங்கட்கிழமை') THEN

day\_column := 'monday\_yogam';

ELSIF day\_name IN ('tue', 'tuesday', 'செவ்வாய்', 'செவ்வாய்க்கிழமை') THEN

day\_column := 'tuesday\_yogam';

ELSIF day\_name IN ('wed', 'wednesday', 'புதன்', 'புதன்கிழமை') THEN

day\_column := 'wednesday\_yogam';

ELSIF day\_name IN ('thu', 'thursday', 'வியாழன்', 'வியாழக்கிழமை') THEN

day\_column := 'thursday\_yogam';

ELSIF day\_name IN ('fri', 'friday', 'வெள்ளி', 'வெள்ளிக்கிழமை') THEN

day\_column := 'friday\_yogam';

ELSIF day\_name IN ('sat', 'saturday', 'சனி', 'சனிக்கிழமை') THEN

day\_column := 'saturday\_yogam';

ELSE

RETURN NULL;

END IF;

*-- Query for the yogam value*

EXECUTE format('

SELECT %I

FROM nakshatra\_yogam\_mappings

WHERE nakshatra\_name = $1 OR nakshatra\_name\_english = $1

LIMIT 1', day\_column)

INTO yogam

USING normalized\_name;

RETURN yogam;

END;

$$ LANGUAGE plpgsql;

**Frontend Implementation**

In the TamilJyotish Daily Panchangam app, the nakshatra yogam is displayed as part of the main panchangam information. The code fetches this data as follows:

javascript

const fetchPanchangamData = async (date) => {

setLoading(true);

*// Format date for API*

const formattedDate = date.toISOString().split('T')[0];

*// Fetch panchangam data*

const { data, error } = await supabase

.from('daily\_panchangam')

.select('\*')

.eq('date', formattedDate)

.single();

if (error || !data) {

*// Error handling...*

return;

}

*// Get nakshatra yogam*

const dayOfWeek = date.toLocaleDateString('en-US', { weekday: 'long' });

try {

const { data: yogamData, error: yogamError } = await supabase

.rpc('get\_nakshatra\_yogam', {

nakshatra\_name: data.main\_nakshatra,

day\_name: dayOfWeek

});

if (yogamError) throw yogamError;

setPanchangamData({...data, nakshatra\_yogam: yogamData});

} catch (e) {

console.error('Error fetching nakshatra yogam:', e);

setPanchangamData(data); *// Set without yogam data*

}

setLoading(false);

};

The nakshatra yogam is rendered in the UI:

jsx

<div className="info-item">

<span className="label">🔮 Nakshatra Yogam: </span>

{panchangamData.nakshatra\_yogam || 'N/A'}

</div>

**Known Issues and Solutions**

**1. Nakshatra Name Spelling Variations**

**Issue**: The nakshatra "Hastham" has two different spellings in Tamil: "அஸ்தம்" and "ஹஸ்தம்". The database query would fail to match these as the same nakshatra.

**Solution**: Two approaches were implemented:

1. **Database Function Normalization**: The RPC function normalizes known variations before lookup

sql

IF nakshatra\_name = 'அஸ்தம்' THEN

normalized\_name := 'ஹஸ்தம்';

ELSE

normalized\_name := nakshatra\_name;

END IF;

1. **Alternative Spelling Entry**: Added both spellings to the mapping table

sql

INSERT INTO nakshatra\_yogam\_mappings (

nakshatra\_name, nakshatra\_name\_english,

*/\* yogam fields \*/*

) VALUES (

'அஸ்தம்', 'Hastham',

*/\* yogam values \*/*

);

**2. Day Name Format Inconsistencies**

**Issue**: Different formats for day names could cause lookups to fail.

**Solution**: The RPC function handles multiple variations of day names in both English and Tamil:

sql

IF day\_name IN ('sun', 'sunday', 'ஞாயிறு', 'ஞாயிற்றுக்கிழமை') THEN

day\_column := 'sunday\_yogam';

*-- Other days similarly handled*

**Complete Mapping Table**

Here's an example of the complete mapping table for all 27 nakshatras (sample values):

| **Nakshatra (Tamil)** | **Nakshatra (English)** | **Sunday** | **Monday** | **Tuesday** | **Wednesday** | **Thursday** | **Friday** | **Saturday** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| அசுவினி | Ashwini | சித்தயோகம் | சித்தயோகம் | சித்தயோகம் | மரணயோகம் | அமிர்தயோகம் | அமிர்தயோகம் | சித்தயோகம் |
| பரணி | Bharani | சித்தயோகம் | சித்தயோகம் | சித்தயோகம் | சித்தயோகம் | சித்தயோகம் | சித்தயோகம் | சித்தயோகம் |
| கார்த்திகை | Krittika | சித்தயோகம் | மரணயோகம் | சித்தயோகம் | அமிர்தயோகம் | மரணயோகம் | சித்தயோகம் | அமிர்தயோகம் |
| ரோகிணி | Rohini | சித்தயோகம் | அமிர்தயோகம் | அமிர்தயோகம் | சித்தயோகம் | மரணயோகம் | மரணயோகம் | அமிர்தயோகம் |
| ஹஸ்தம்/அஸ்தம் | Hastham | அமிர்தயோகம் | சித்தயோகம் | சித்தயோகம் | மரணயோகம் | சித்தயோகம் | அமிர்தயோகம் | மரணயோகம் |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |

**Astrological Significance**

In Tamil astrology, the nakshatra yogam provides guidance on:

1. **Timing of Activities**: When to perform important tasks for the best outcome
2. **Decision Making**: Favorable or unfavorable days for major decisions
3. **Spiritual Practices**: Days that enhance the effect of specific spiritual activities
4. **Travel Plans**: Determining good days for travel based on one's birth star

**Best Practices for Maintenance**

1. **Adding New Nakshatras**:

sql

INSERT INTO nakshatra\_yogam\_mappings (

nakshatra\_name, nakshatra\_name\_english,

sunday\_yogam, monday\_yogam, tuesday\_yogam, wednesday\_yogam,

thursday\_yogam, friday\_yogam, saturday\_yogam

) VALUES (

'நக்ஷத்திரம்', 'New Nakshatra',

'சித்தயோகம்', 'அமிர்தயோகம்', 'மரணயோகம்', 'சித்தயோகம்',

'அமிர்தயோகம்', 'சித்தயோகம்', 'மரணயோகம்'

);

1. **Handling Spelling Variations**:
   * Add both spellings to the mapping table, or
   * Update the RPC function to normalize known variations
2. **Verifying Data Integrity**:

sql

*-- Check for missing nakshatras*

SELECT DISTINCT main\_nakshatra

FROM daily\_panchangam

WHERE main\_nakshatra NOT IN (

SELECT nakshatra\_name FROM nakshatra\_yogam\_mappings

);

This documentation covers the implementation, known issues, and maintenance of the Amruthathi Yogam system in the TamilJyotish Daily Panchangam application.