SQL – User\_forecast\_calendar creation

CREATE TABLE user\_forecast\_calendar (

calendar\_id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

user\_id UUID NOT NULL,

reference\_id UUID REFERENCES user\_income\_expense\_schedule(id),

due\_date DATE NOT NULL,

amount NUMERIC NOT NULL,

event\_type TEXT NOT NULL CHECK (event\_type IN ('income', 'expense')),

created\_at TIMESTAMP DEFAULT now()

);

once the dates are generated, what table or file do you put them in?

**ChatGPT said:**

Excellent question — once the dates are generated by your Python logic (generate\_schedule.py), they are inserted into the **forecast\_calendar** table in your **Supabase database**.

**📊 Target Table: forecast\_calendar**

This table is designed to store **every individual future income or expense event** that your app will display or analyze.

**✅ Example Use Cases:**

* Daily cash balance forecasting
* Alerts for when cash might run low
* Charts of future income vs. bills
* Calendar views in your mobile app

**🗃 Recommended Table Schema for forecast\_calendar**

sql

Copy code

CREATE TABLE forecast\_calendar (

calendar\_id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

user\_id UUID NOT NULL,

reference\_id UUID, -- Links to the income\_expense\_schedule entry

due\_date DATE NOT NULL,

amount NUMERIC NOT NULL,

event\_type TEXT CHECK (event\_type IN ('income', 'expense')),

created\_at TIMESTAMP DEFAULT now()

);

**📥 How the Data Gets Into This Table**

When you run:

python

Copy code

insert\_forecast\_dates(

user\_id="abc123",

reference\_id="schedule456",

name="Paycheck",

frequency="bi-weekly",

start\_date="2025-06-01",

amount=1500,

event\_type="income"

)

The forecast\_loader.py script:

1. Uses generate\_schedule() to return a list like:

python

Copy code

["2025-06-01", "2025-06-15", "2025-06-29"]

1. Builds a set of row entries:

json

Copy code

{

"user\_id": "abc123",

"reference\_id": "schedule456",

"due\_date": "2025-06-01",

"amount": 1500,

"event\_type": "income"

}

1. Sends those to Supabase:

python

Copy code

supabase.table("forecast\_calendar").insert(inserts).execute()

**🧠 Why This Structure Works Well**

* Easy to query: "Show all upcoming events for user X"
* Easy to chart: "Sum income vs. expenses for each day"
* Easy to link back to the original schedule
* Flexible for future features (alerts, repeats, skipping)