Stakeholder GTM questions (Payroll SaaS)

Sales:
Do we have enough pipeline to hit next quarter’s ARR target by segment?
What's the win rate by segment and by primary competitor?
How long is the sales cycle by segment and channel (inbound vs outbound vs partner)?
Are we discounting more in certain segments, and does it improve close rate or just lower ARR?
Which reps perform best by segment (win rate + cycle time + ACV), and where do deals stall by stage?
Marketing:
Which Campaigns (or) Channels generate the highest Marketing Qualified Lead (MQL) -> Sales Qualified Lead(SQL) → Opportunity conversion in each segment?
What is the cost per opportunity and cost per won deal in each segment?
Which channels create fastest conversion pipelines in each segment?
Customer Success:
What is the time-to-go-live in each segment and point which segment has the longest time-to-go-live?
What is the gross retention and net retention in each segment and where is churn?
Number of tickets per 100 employees by segment? Do high severity tickets predict churn?
Product
Which product is notable to predict expansion by segment?
Which segments adopt most often and what is the increase in ARR?
Which segments saw an increase in retention when a module is enabled?
Leadership
See the Annual Recurring Revenue distribution and growth by segment. Are we reliant on small and medium businesses or enterprises?

Step by Step implementation of the project:

Tech stack: BigQuery + DBT + CI/CD in DBT + Git + SQL

Step 1: Setting up Bigquery

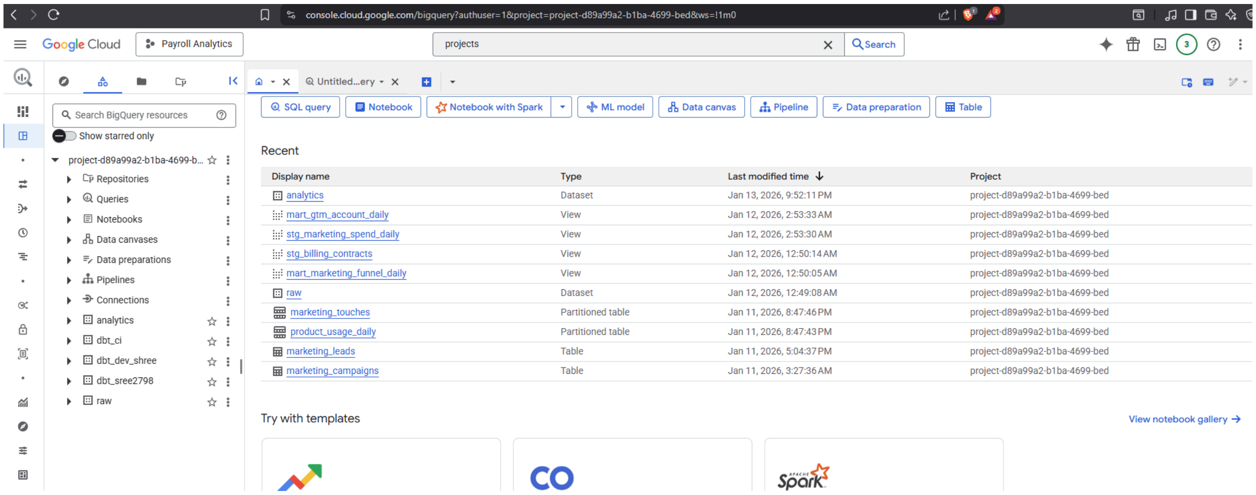
1) Create a GCP project and create datasets
raw (to store source tables, source tables are generated by SQL script here. But, in realtime, the source tables will be landed in raw layer via connectors like Fivetran (native connectors or custom connectors to sync the data via REST API or GraphQL API)

Analytics (business stakeholders consume data from this layer, generally this layer is connected to a visualization tool like looker or power bi)

dbt_sree_2798 (this schema contains the tables that will be populated with the sql transformations made in my personal development branch)

dbt_ci (this schema is for CI purpose)

Once the setup is completed, I can see the GCP UI as below



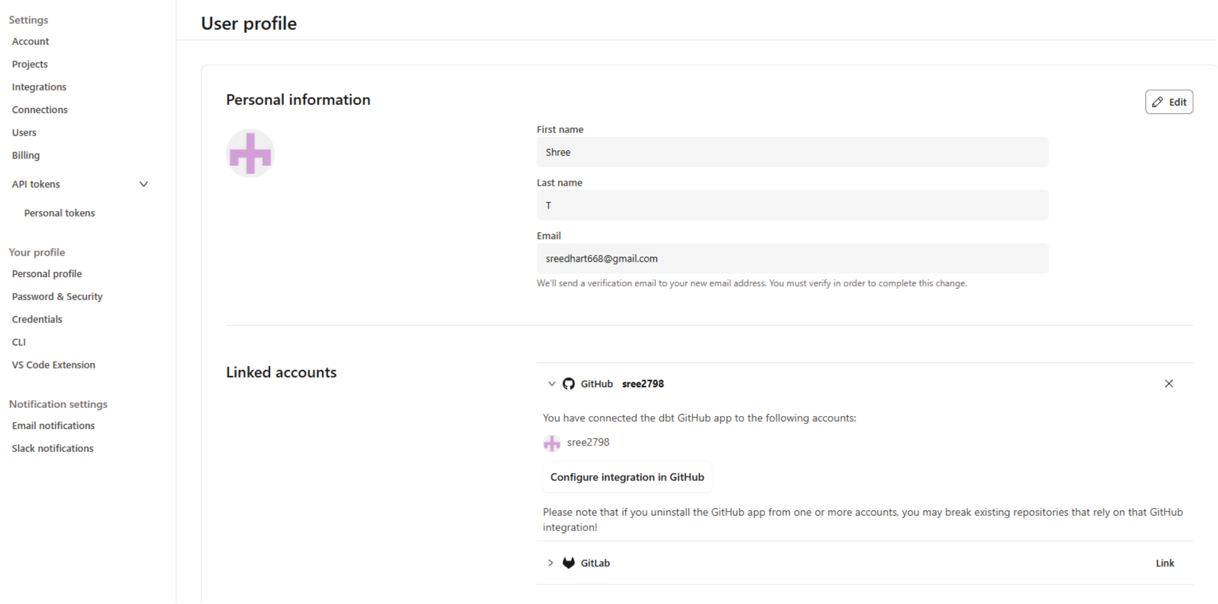
Recent

Display name	Type	Last modified time ↓	Project
analytics	Dataset	Jan 13, 2026, 9:52:11 PM	project-d89a99a2-b1ba-4699-bed
mart_gtm_account_daily	View	Jan 12, 2026, 2:53:33 AM	project-d89a99a2-b1ba-4699-bed
stg_marketing_spend_daily	View	Jan 12, 2026, 2:53:30 AM	project-d89a99a2-b1ba-4699-bed
stg_billing_contracts	View	Jan 12, 2026, 12:50:14 AM	project-d89a99a2-b1ba-4699-bed
mart_marketing_funnel_daily	View	Jan 12, 2026, 12:50:05 AM	project-d89a99a2-b1ba-4699-bed
raw	Dataset	Jan 12, 2026, 12:49:08 AM	project-d89a99a2-b1ba-4699-bed
marketing_touches	Partitioned table	Jan 11, 2026, 8:47:46 PM	project-d89a99a2-b1ba-4699-bed
product_usage_daily	Partitioned table	Jan 11, 2026, 8:47:43 PM	project-d89a99a2-b1ba-4699-bed
marketing_leads	Table	Jan 11, 2026, 5:04:37 PM	project-d89a99a2-b1ba-4699-bed
marketing_campaigns	Table	Jan 11, 2026, 3:27:36 AM	project-d89a99a2-b1ba-4699-bed

Step 2: DBT cloud + git integration

Created a github repo gtm-payroll-analytics engineering (Link: Payroll Analytics)
Created a project in DBT cloud and connected the project to github repo

DBT cloud should be able to write data in Big query. So, I created a service account in BigQuery and downloaded the JSON file to import the same in DBT cloud and I set the default dataset to 'dbt_sree2798'



User profile

Personal information

First name: Shree

Last name: T

Email: sreedhart668@gmail.com

We'll send a verification email to your new email address. You must verify in order to complete this change.

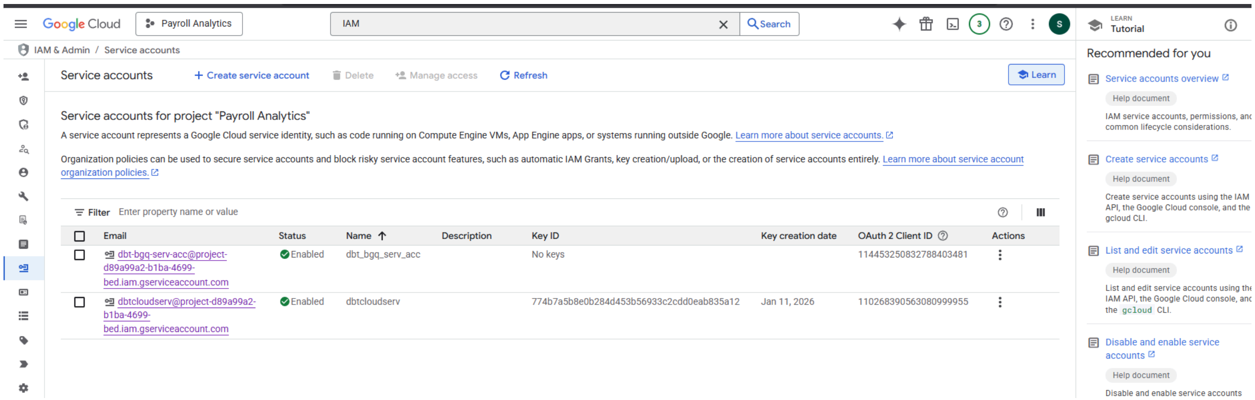
Linked accounts

GitHub sree2798

You have connected the dbt GitHub app to the following accounts:

sree2798

Please note that if you uninstall the GitHub app from one or more accounts, you may break existing repositories that rely on that GitHub integration!



Service accounts for project "Payroll Analytics"

A service account represents a Google Cloud service identity, such as code running on Compute Engine VMs, App Engine apps, or systems running outside Google. Learn more about service accounts.

Organization policies can be used to secure service accounts and block risky service account features, such as automatic IAM Grants, key creation/upload, or the creation of service accounts entirely. Learn more about service account organization policies.

Filter Enter property name or value

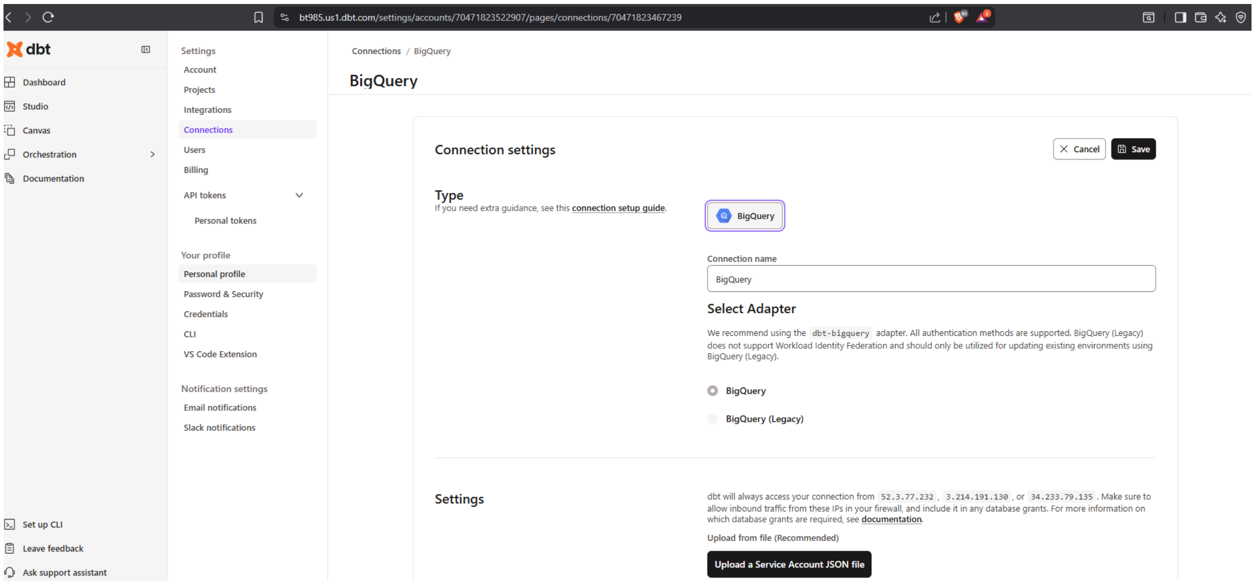
□	Email	Status	Name ↑	Description	Key ID	Key creation date	OAuth 2 Client ID	Actions
□	dbt-bgq-serv-acc@project-d89a99a2-b1ba-4699-bed.iam.gserviceaccount.com	✓ Enabled	dbt_bgq_serv_acc		No keys		114453250832788403481	⋮
□	dbtcloudserv@project-d89a99a2-b1ba-4699-bed.iam.gserviceaccount.com	✓ Enabled	dbtcloudserv		774b7a5b8e0b284d453b56933c2cdd0eab835a12	Jan 11, 2026	110268390563080999955	⋮

IAM service accounts, permissions, and common lifecycle considerations.

Create service accounts using the IAM API, the Google Cloud console, and the gcloud CLI.

List and edit service accounts using the IAM API, the Google Cloud console, and the `gcloud` CLI.

Disable and enable service accounts



BigQuery

Connection settings

Type
If you need extra guidance, see this connection setup guide.

Connection name: BigQuery

Select Adapter

We recommend using the `dbt-bigquery` adapter. All authentication methods are supported. BigQuery (Legacy) does not support Workload Identity Federation and should only be utilized for updating existing environments using BigQuery (Legacy).

● BigQuery
○ BigQuery (Legacy)

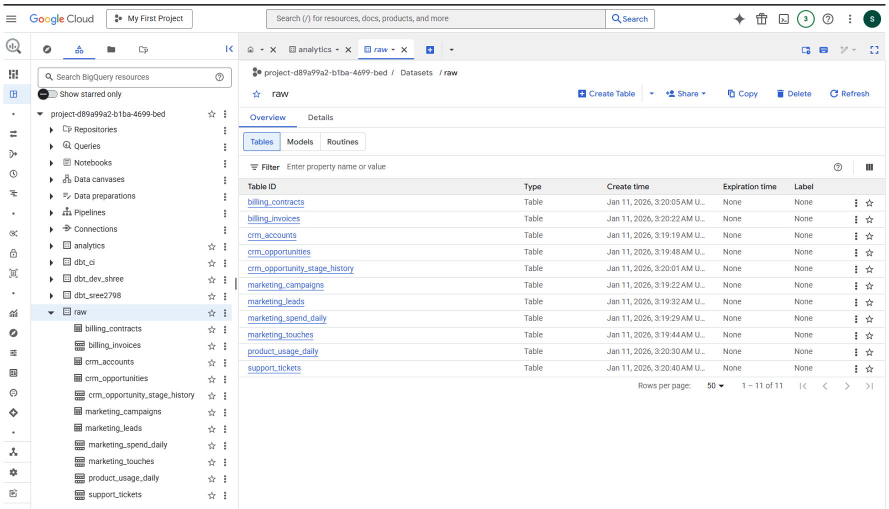
Settings

dbt will always access your connection from `52.3.77.232`, `3.214.191.130`, or `34.233.79.135`. Make sure to allow inbound traffic from these IPs in your firewall, and include it in any database grants. For more information on which database grants are required, see documentation.

Upload from file (Recommended)

Step 3: Generate raw data with SQL script

Refer to the SQL script in github repo here (generate_raw_data)



project-d89a99a2-b1ba-4699-bed / Datasets / raw

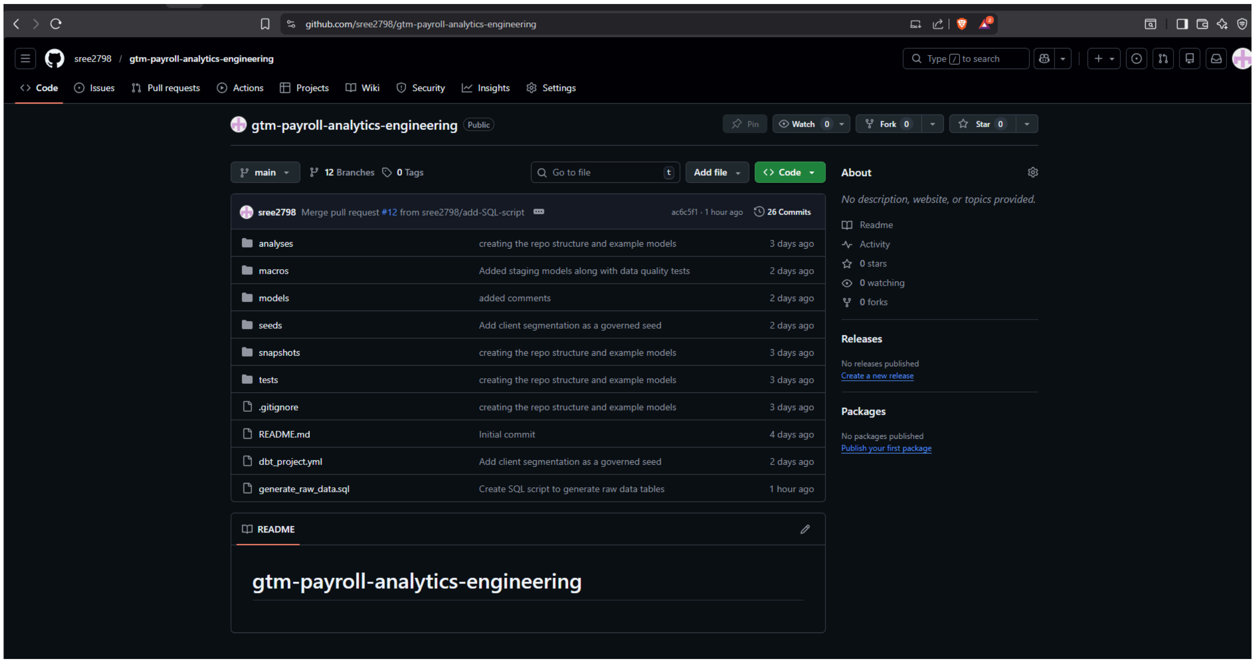
raw

Filter Enter property name or value

Table ID	Type	Create time	Expiration time	Label
billing_contracts	Table	Jan 11, 2026, 3:20:05 AM U...	None	None
billing_invoices	Table	Jan 11, 2026, 3:20:22 AM U...	None	None
crm_accounts	Table	Jan 11, 2026, 3:19:19 AM U...	None	None
crm_opportunities	Table	Jan 11, 2026, 3:19:48 AM U...	None	None
crm_opportunity_stage_history	Table	Jan 11, 2026, 3:20:01 AM U...	None	None
marketing_campaigns	Table	Jan 11, 2026, 3:19:22 AM U...	None	None
marketing_leads	Table	Jan 11, 2026, 3:19:32 AM U...	None	None
marketing_spend_daily	Table	Jan 11, 2026, 3:19:29 AM U...	None	None
marketing_touches	Table	Jan 11, 2026, 3:19:44 AM U...	None	None
product_usage_daily	Table	Jan 11, 2026, 3:20:30 AM U...	None	None
support_tickets	Table	Jan 11, 2026, 3:20:40 AM U...	None	None

Rows per page: 50 1 – 11 of 11

The github repo will contain a structure as below



github.com/sree2798/gtm-payroll-analytics-engineering

sree2798 / gtm-payroll-analytics-engineering

gtm-payroll-analytics-engineering Public

main · 12 Branches · 0 Tags

sree2798 Merge pull request #12 from sree2798/add-SQL-script · ac6c5f1 · 1 hour ago · 26 Commits

analyses	creating the repo structure and example models	3 days ago
macros	Added staging models along with data quality tests	2 days ago
models	added comments	2 days ago
seeds	Add client segmentation as a governed seed	2 days ago
snapshots	creating the repo structure and example models	3 days ago
tests	creating the repo structure and example models	3 days ago
.gitignore	creating the repo structure and example models	3 days ago
README.md	Initial commit	4 days ago
dbt_project.yml	Add client segmentation as a governed seed	2 days ago
generate_raw_data.sql	Create SQL script to generate raw data tables	1 hour ago

README

gtm-payroll-analytics-engineering

About

No description, website, or topics provided.

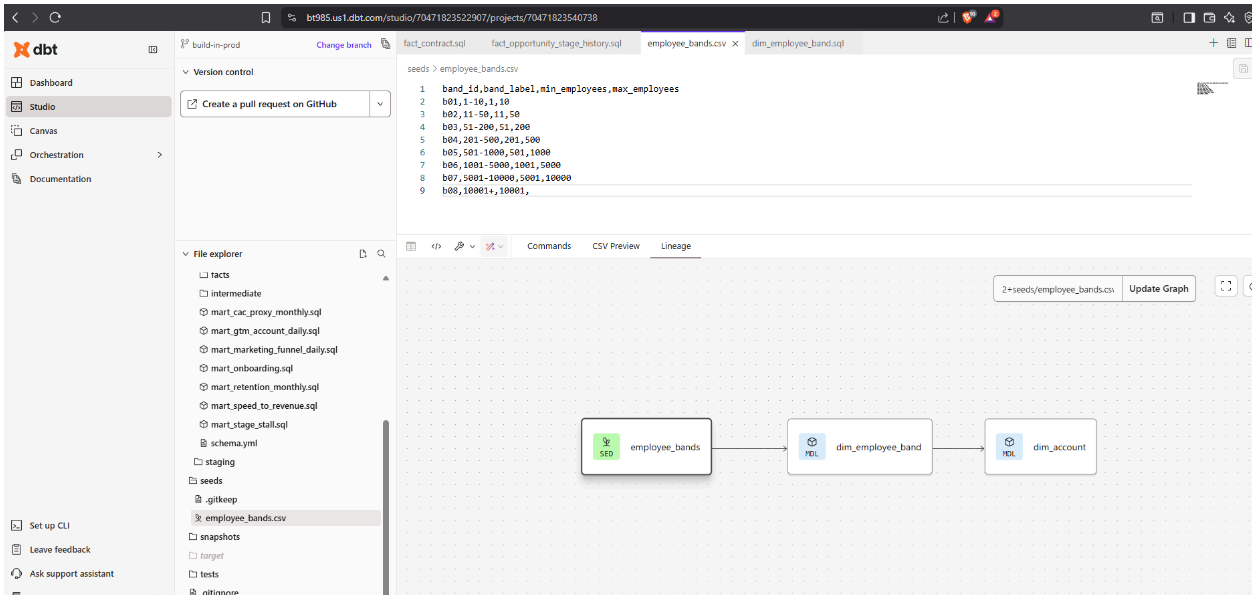
Releases

No releases published

Packages

No packages published

Step 5: Include client segmentation as a seed

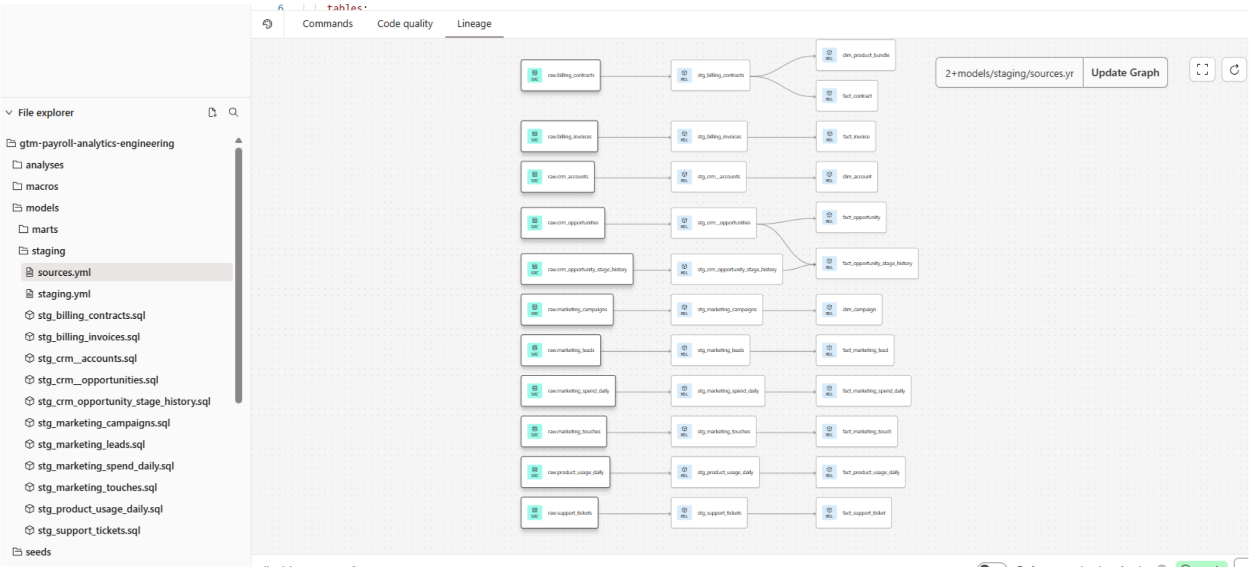


bt985.us1.dbt.com/studio/70471823522907/projects/70471823540738

seeds > employee_bands.csv

```
band_id,band_label,min_employees,max_employees
b01,1-10,1,10
b02,11-50,11,50
b03,51-200,51,200
b04,201-500,201,500
b05,501-1000,501,1000
b06,1001-5000,1001,5000
b07,5001-10000,5001,10000
b08,10001+,10001,
```

Step 6: Add sources to DBT which point to the raw tables created by the SQL script



```
6     tables:
```

Step 7: Build staging models

Build one staging model per raw table

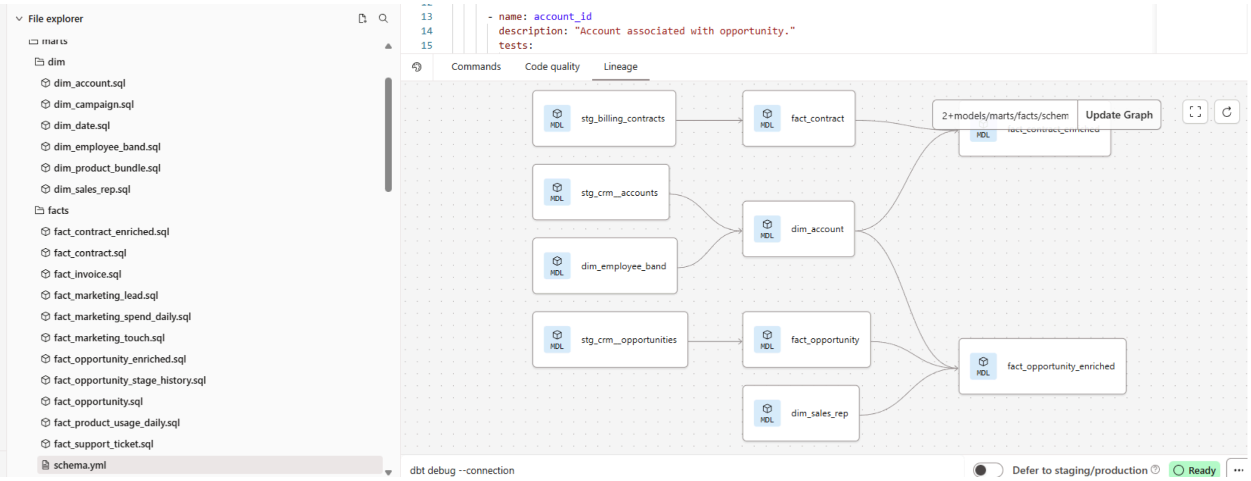
Each staging model cast the data types, standardizes the naming and deduplicates if needed

Also, add tests in staging.yml, this file contains

In realtime environment, to optimize build time in DBT, we can implement staging models only to the tables that are actually required to create required dimensions and facts

Also, we include only the required columns in staging layer to build the metrics

Step 9: Build dimensions and facts:



```
File explorer
  marts
    dim
      dim_account.sql
      dim_campaign.sql
      dim_date.sql
      dim_employee_band.sql
      dim_product_bundle.sql
      dim_sales_rep.sql
    facts
      fact_contract_enriched.sql
      fact_contract.sql
      fact_invoice.sql
      fact_marketing_lead.sql
      fact_marketing_spend_daily.sql
      fact_marketing_touch.sql
      fact_opportunity_enriched.sql
      fact_opportunity_stage_history.sql
      fact_opportunity.sql
      fact_product_usage_daily.sql
      fact_support_ticket.sql
      schema.yml
```

```
- name: account_id
  description: "Account associated with opportunity."
  tests:
```

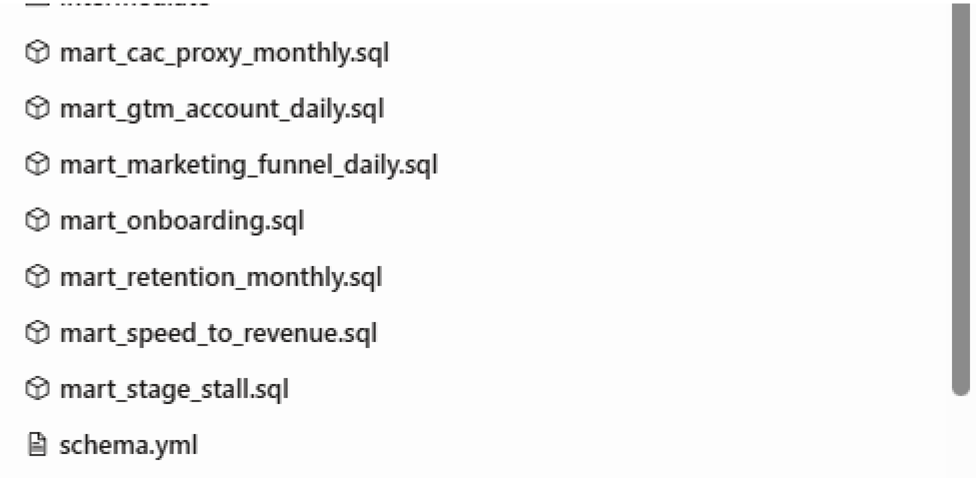
```
dbt debug --connection
```

Step 10:

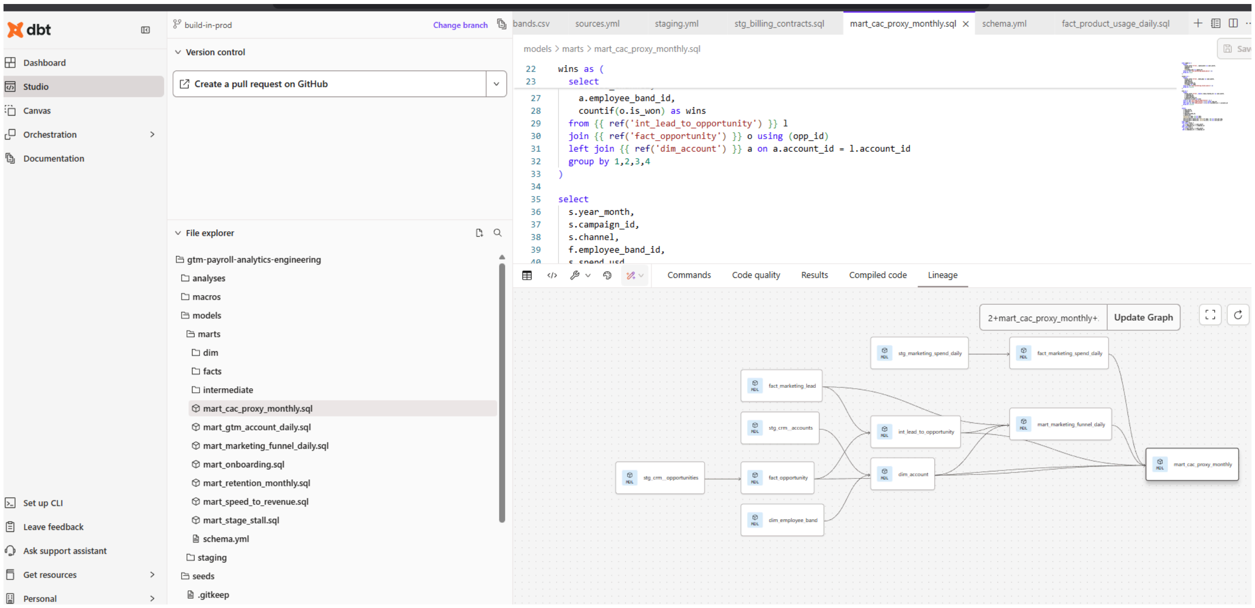
Build the gold layer to answer stakeholder questions

Create mart_gtm_account_daily with a grain of one row per account per day

With current ARR, daily rollups, rolling 30 day payroll runs, rolling 30 day ticket counts and segment



```
mart_cac_proxy_monthly.sql
mart_gtm_account_daily.sql
mart_marketing_funnel_daily.sql
mart_onboarding.sql
mart_retention_monthly.sql
mart_speed_to_revenue.sql
mart_stage_stall.sql
schema.yml
```



```
wins as (
  select
    a.employee_band_id,
    countif(o.is_won) as wins
  from {{ ref('int_lead_to_opportunity') }} l
  join {{ ref('fact_opportunity') }} o using (opp_id)
  left join {{ ref('dim_account') }} a on a.account_id = l.account_id
  group by 1,2,3,4
)

select
  s.year_month,
  s.campaign_id,
  s.channel,
  f.employee_band_id,
  s.spend_usd
```

Step 11: CI/CD in DBT cloud

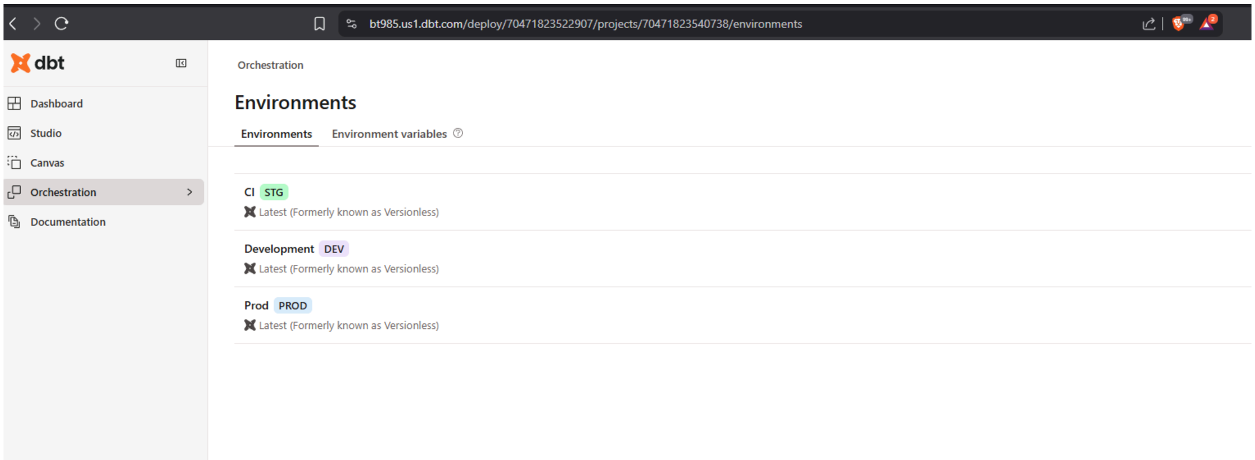
Create 3 environments in DBT cloud.

Dev → dbt_sree2798 (models built in dev will be visible in the given schema)
Dbt_ci → This schema is for CI purpose
Prod → The schema ‘analytics’ will contain the tables that are built in prod environment (consumption layer)

Create 2 jobs

PR CI job → This job will be triggered on a pull request (dbt build)

Prod deploy job → This job will be triggered when branch merges to main


bt985.us1.dbt.com/deploy/70471823522907/projects/70471823540738/environments
dbt
Dashboard
Studio
Canvas
Orchestration
Documentation

Orchestration

Environments

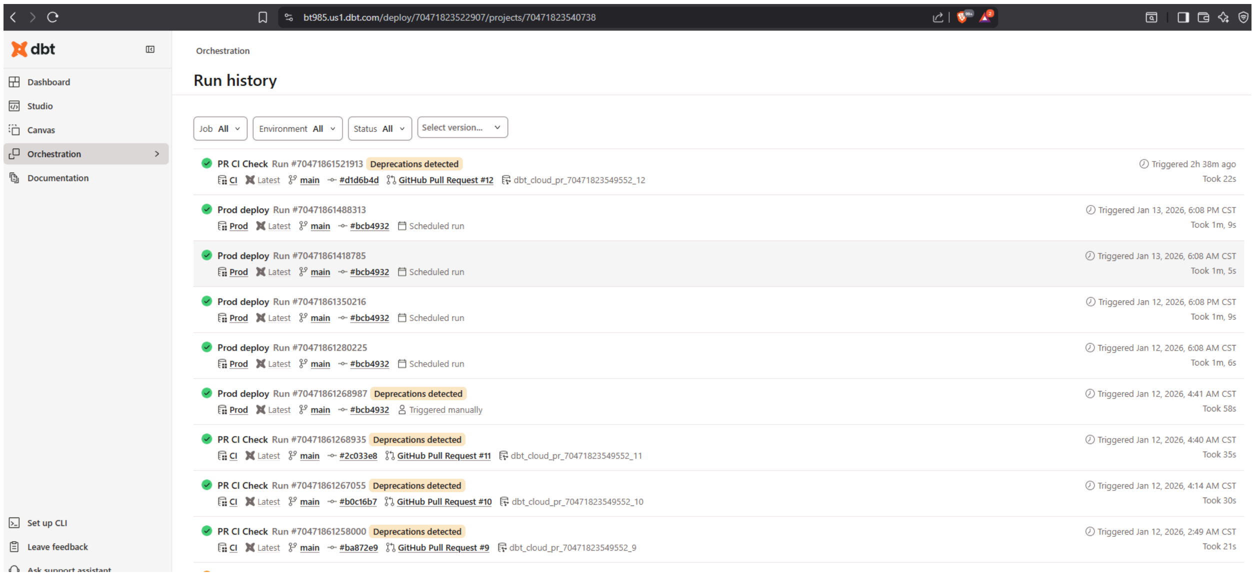
Environments | Environment variables

CI STG
Latest (Formerly known as Versionless)

Development DEV
Latest (Formerly known as Versionless)

Prod PROD
Latest (Formerly known as Versionless)

Few sample runs triggered during PRs in github



bt985.us1.dbt.com/deploy/70471823522907/projects/70471823540738

dbt

Dashboard
Studio
Canvas
Orchestration
Documentation

Set up CLI
Leave feedback
Ask support assistant

Orchestration

Run history

Job All | Environment All | Status All | Select version...

PR CI Check Run #70471861521913 Deprecations detected — Triggered 2h 38m ago
CI Latest main #d1d6b4d GitHub Pull Request #12 dbt_cloud_pr_70471823549552_12 — Took 22s

Prod deploy Run #70471861488313 — Triggered Jan 13, 2026, 6:08 PM CST
Prod Latest main #bcb4932 Scheduled run — Took 1m, 9s

Prod deploy Run #70471861418785 — Triggered Jan 13, 2026, 6:08 AM CST
Prod Latest main #bcb4932 Scheduled run — Took 1m, 5s

Prod deploy Run #70471861350216 — Triggered Jan 12, 2026, 6:08 PM CST
Prod Latest main #bcb4932 Scheduled run — Took 1m, 9s

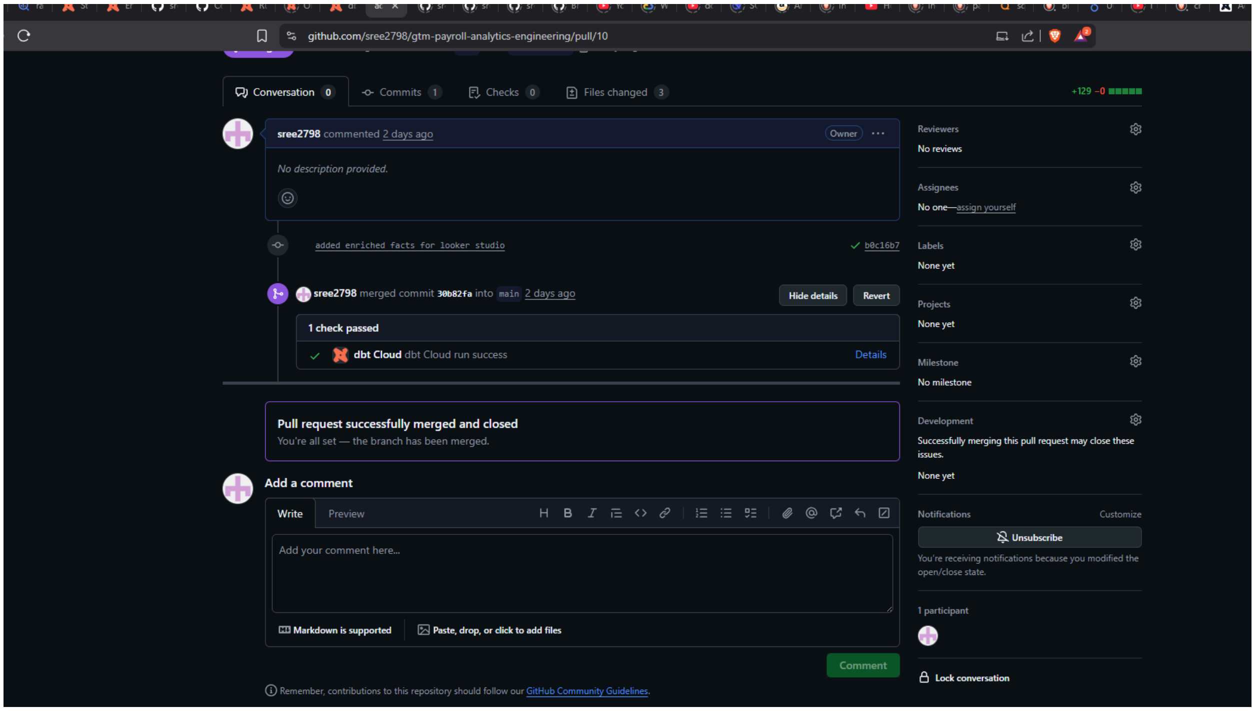
Prod deploy Run #70471861280225 — Triggered Jan 12, 2026, 6:08 AM CST
Prod Latest main #bcb4932 Scheduled run — Took 1m, 6s

Prod deploy Run #70471861268987 Deprecations detected — Triggered Jan 12, 2026, 4:41 AM CST
Prod Latest main #bcb4932 Triggered manually — Took 58s

PR CI Check Run #70471861268935 Deprecations detected — Triggered Jan 12, 2026, 4:40 AM CST
CI Latest main #2c033e8 GitHub Pull Request #11 dbt_cloud_pr_70471823549552_11 — Took 35s

PR CI Check Run #70471861267055 Deprecations detected — Triggered Jan 12, 2026, 4:14 AM CST
CI Latest main #b0c16b7 GitHub Pull Request #10 dbt_cloud_pr_70471823549552_10 — Took 30s

PR CI Check Run #70471861258000 Deprecations detected — Triggered Jan 12, 2026, 2:49 AM CST
CI Latest main #ba872e9 GitHub Pull Request #9 dbt_cloud_pr_70471823549552_9 — Took 21s



github.com/sree2798/gtm-payroll-analytics-engineering/pull/10

Conversation 0 | Commits 1 | Checks 0 | Files changed 3 — +129 −0

sree2798 commented 2 days ago — Owner

No description provided.

`added enriched facts for looker studio` ✓ b0c16b7

sree2798 merged commit **30b82fa** into `main` 2 days ago — Hide details | Revert

1 check passed

✓ **dbt Cloud** dbt Cloud run success — Details

Pull request successfully merged and closed

You're all set — the branch has been merged.

Add a comment

Write | Preview

Add your comment here...

Markdown is supported | Paste, drop, or click to add files

Comment

Remember, contributions to this repository should follow our GitHub Community Guidelines.

Reviewers
No reviews

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Development
Successfully merging this pull request may close these issues.
None yet

Notifications — Customize
Unsubscribe
You're receiving notifications because you modified the open/close state.

1 participant

Lock conversation