

Database Setup Guide

This guide covers database configuration and schema setup for the BrazilRetail-BI project.

Prerequisites

- PostgreSQL 12+
- Database user with creation privileges

Step 1: Create Database

Connect to PostgreSQL and create the database:

```
CREATE DATABASE brazilretail_bi;
```

Step 2: Create Database User (Optional)

Create a dedicated user for the application:

```
CREATE USER brazilretail_user WITH PASSWORD 'your_password';
GRANT ALL PRIVILEGES ON DATABASE brazilretail_bi TO
brazilretail_user;
```

Step 3: Configure Environment

Update the `.env` file in the project root:

```
# Local PostgreSQL

DATABASE_URL=postgresql://username:password@localhost:5432/brazilretail_bi
# Production Supabase
SUPABASE_URL=https://your-project.supabase.co
SUPERKEY=your-service-role-key
```

Replace with your actual credentials.

Step 4: Create Schema

Run the schema creation script (works for both Local and Supabase if configured):

```
python db_schema/create_schema.py
```

This creates all tables:

- customers
- geolocation
- orders
- order_items
- order_payments
- order_reviews
- products
- sellers

Step 5: Setup Constraints and Indexes

After creating the schema, run the constraints script to add Foreign Keys and Indexes. This is crucial for performance and data integrity.

```
python db_schema/setup_constraints.py
```

Database Schema Overview

Core Tables

customers

- customer_id (PK)
- customer_unique_id
- customer_zip_code_prefix
- customer_city
- customer_state
- customer_state_initials

orders

- order_id (PK)
- customer_id (FK)
- order_status
- order_purchase_timestamp
- order_approved_at
- order_delivered_carrier_date
- order_delivered_customer_date
- order_estimated_delivery_date

order_items

- order_item_id (PK)
- order_id (FK)
- product_id (FK)
- seller_id (FK)
- shipping_limit_date
- price
- freight_value

products

- product_id (PK)
- product_category_name
- product_category_name_english
- product_name_length
- product_description_length
- product_photos_qty
- product_weight_g
- product_length_cm
- product_height_cm
- product_width_cm

Supporting Tables

sellers

- seller_id (PK)
- seller_zip_code_prefix
- seller_city
- seller_state
- seller_state_initials

order_payments

- order_id (PK)
- payment_sequential (PK)
- payment_type
- payment_installments
- payment_value

order_reviews

- review_id (PK)
- order_id (FK)
- review_score
- review_comment_title
- review_comment_message
- review_creation_date
- review_answer_timestamp

geolocation

- geolocation_zip_code_prefix (PK)
- geolocation_lat
- geolocation_lng
- geolocation_city
- geolocation_state
- geolocation_state_initials

Data Loading

After schema creation, run the ETL pipeline to load data:

```
python etl/main.py
```

The pipeline will extract, transform, and load all datasets into the database.

Verification

Connect to the database and verify data loading:

```
SELECT COUNT(*) FROM customers;
SELECT COUNT(*) FROM orders;
-- Check other tables
```

Backup and Maintenance

Regular backups are recommended:

```
pg_dump brazilretail_bi > brazilretail_backup.sql
```

Troubleshooting

- **Connection errors:** Verify DATABASE_URL format and credentials
- **Permission denied:** Ensure user has proper database privileges
- **Schema creation fails:** Check if tables already exist (drop them first if needed)
- **Data loading errors:** Check CSV files are present and ETL logs for details