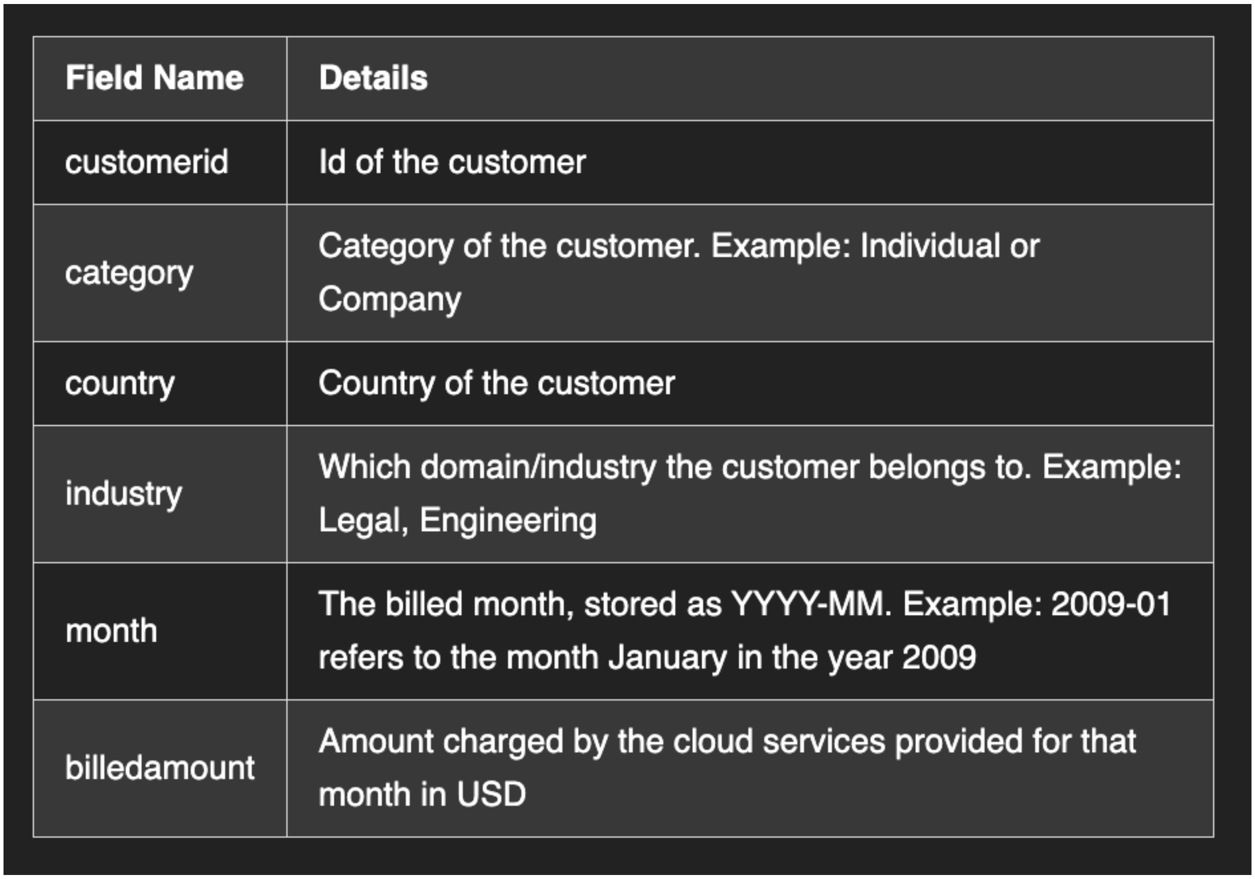
Hands-on Lab: Working with Facts and Dimension Tables

Exercise 1 - Study the schema of the given csv file

In this lab, we will design a data warehouse for a cloud service provider.

- The cloud service provider has given us their billing data in the csv file cloud-billing-dataset.csv.
- This file contains the billing data for the past decade.

Here are the field wise details of the billing data.

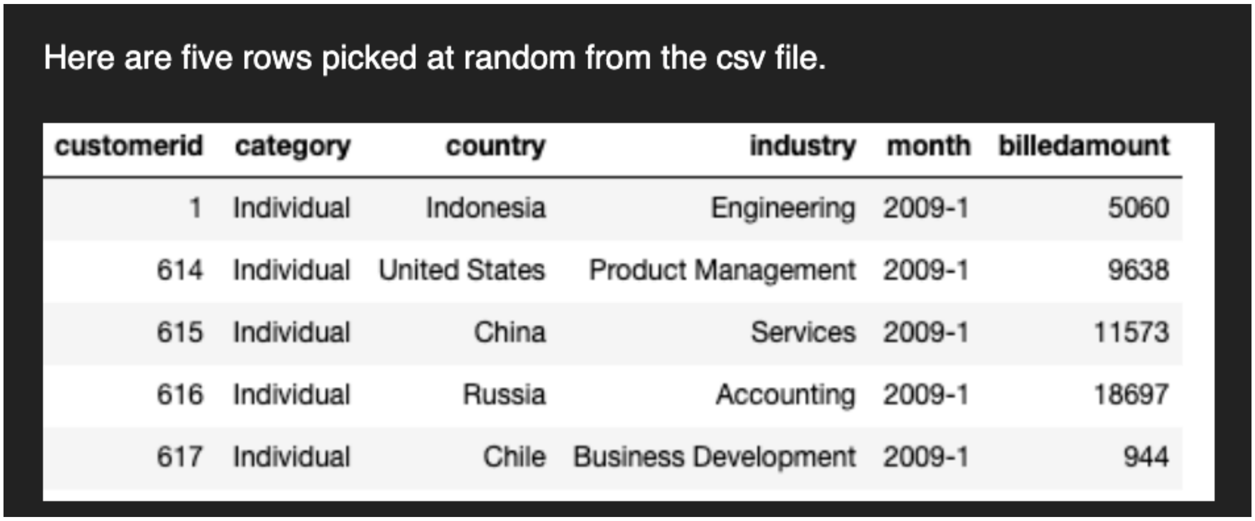


Field Name	Details
customerid	Id of the customer
category	Category of the customer. Example: Individual or Company
country	Country of the customer
industry	Which domain/industry the customer belongs to. Example: Legal, Engineering
month	The billed month, stored as YYYY-MM. Example: 2009-01 refers to the month January in the year 2009
billedamount	Amount charged by the cloud services provided for that month in USD

We need to design a data warehouse that can support the queries listed below:

- average billing per customer
- billing by country
- top 10 customers
- top 10 countries

- billing by industry
- billing by category
- billing by year
- billing by month
- billing by quarter
- average billing per industry per month
- average billing per industry per quarter
- average billing per country per quarter
- average billing per country per industry per quarter



Here are five rows picked at random from the csv file.

customerid	category	country	industry	month	billedamount
1	Individual	Indonesia	Engineering	2009-1	5060
614	Individual	United States	Product Management	2009-1	9638
615	Individual	China	Services	2009-1	11573
616	Individual	Russia	Accounting	2009-1	18697
617	Individual	Chile	Business Development	2009-1	944

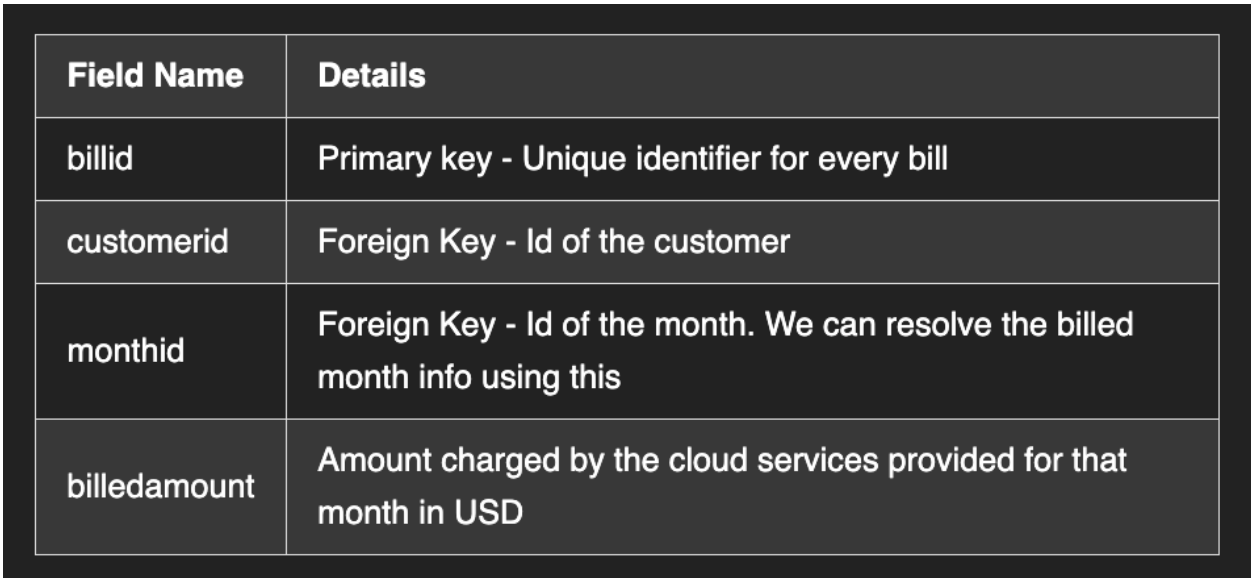
Exercise 2 - Design the fact tables

The fact in this data is the bill which is generated monthly.

The fields **customerid** and **billedamount** are the important fields in the fact table.

We also need a way to identify the additional customer information, other than the id, and date information. So we need fields that refer to the customer and date information in other tables.

The final fact table for the bill would look like this:



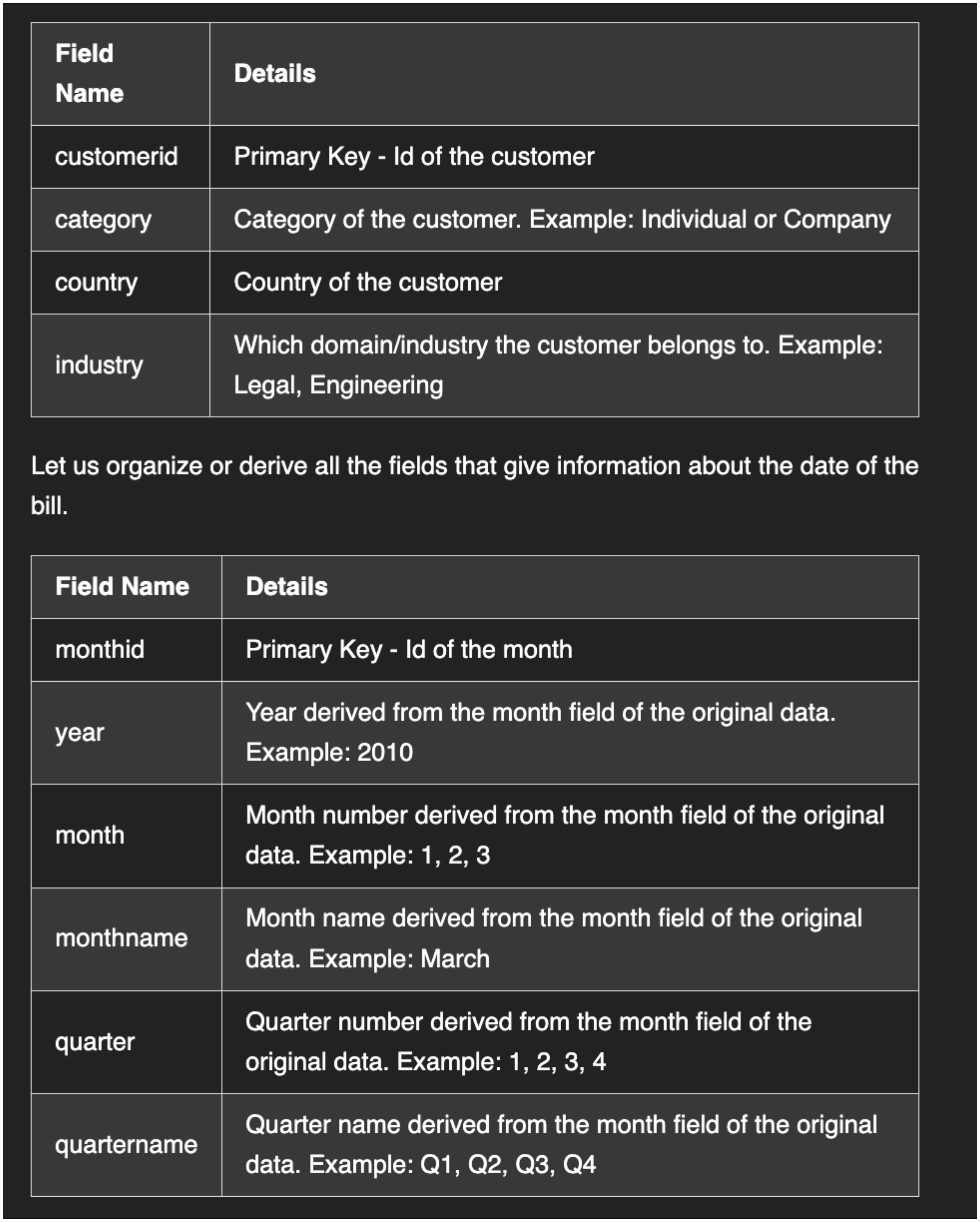
Field Name	Details
billid	Primary key - Unique identifier for every bill
customerid	Foreign Key - Id of the customer
monthid	Foreign Key - Id of the month. We can resolve the billed month info using this
billedamount	Amount charged by the cloud services provided for that month in USD

Exercise 3 - Design the dimension tables

There are two dimensions to our fact(monthly bill).

- Customer information
- Date information

Let us organize all the fields that give information about the customer into a dimension table.



Field Name	Details
customerid	Primary Key - Id of the customer
category	Category of the customer. Example: Individual or Company
country	Country of the customer
industry	Which domain/industry the customer belongs to. Example: Legal, Engineering

Let us organize or derive all the fields that give information about the date of the bill.

Field Name	Details
monthid	Primary Key - Id of the month
year	Year derived from the month field of the original data. Example: 2010
month	Month number derived from the month field of the original data. Example: 1, 2, 3
monthname	Month name derived from the month field of the original data. Example: March
quarter	Quarter number derived from the month field of the original data. Example: 1, 2, 3, 4
quartername	Quarter name derived from the month field of the original data. Example: Q1, Q2, Q3, Q4

Exercise 4 - Create a star schema using the fact and dimension tables

- Based on the previous two exercises, we have now arrived at 3 tables, we can name them as in the table below.

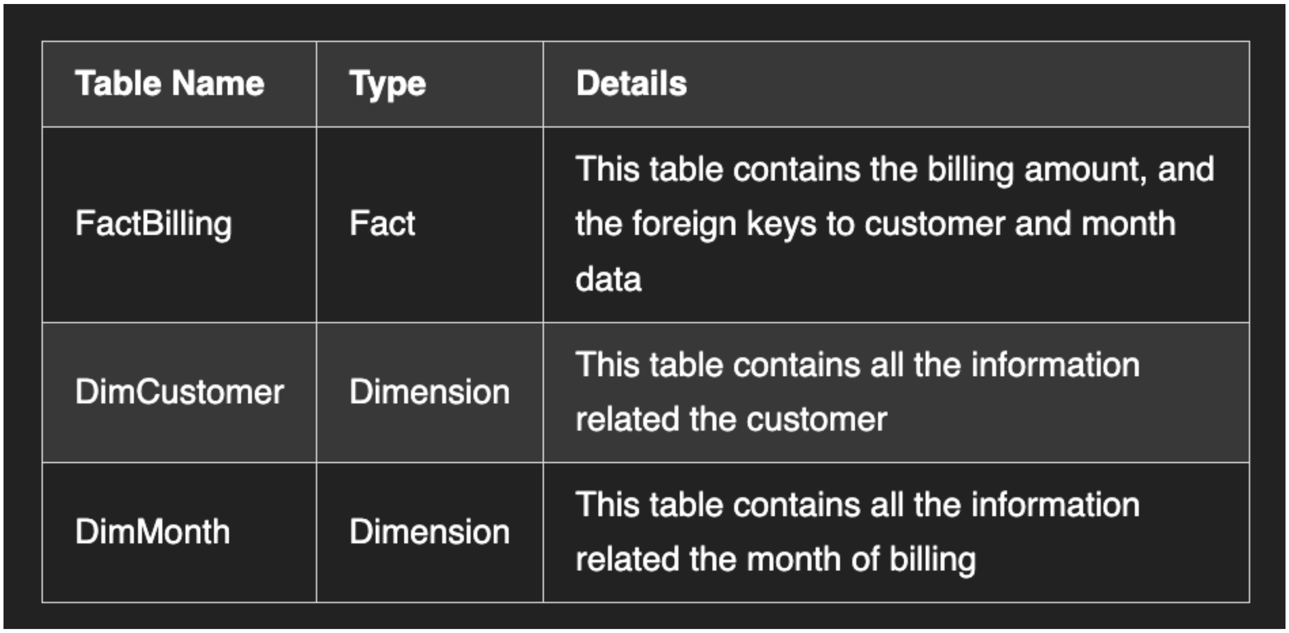
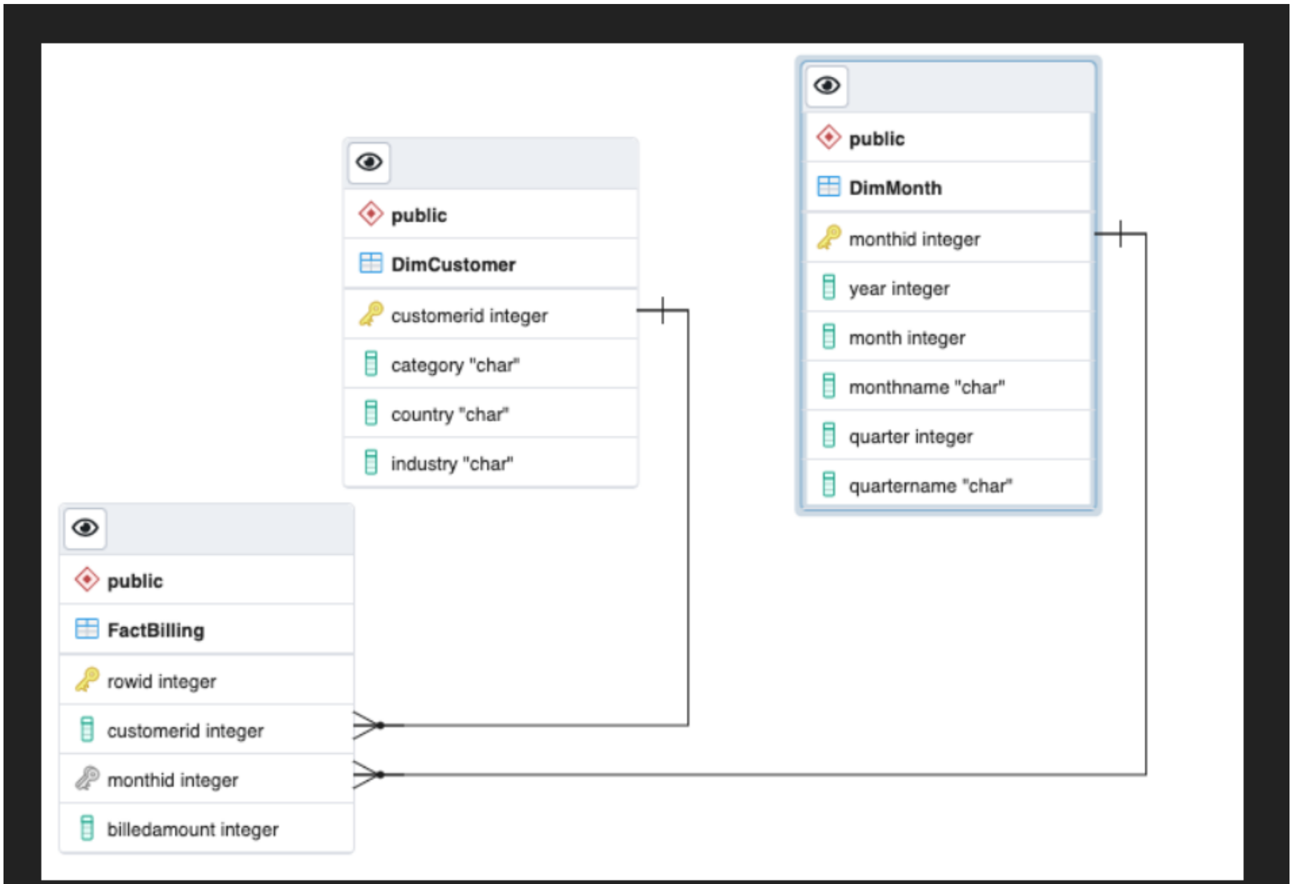


Table Name	Type	Details
FactBilling	Fact	This table contains the billing amount, and the foreign keys to customer and month data
DimCustomer	Dimension	This table contains all the information related the customer
DimMonth	Dimension	This table contains all the information related the month of billing

When we arrange the above tables in Star Schema style, we get a table structure that looks like the one in the image below.



The image shows the fact and dimension tables along with the relaionships between them.

Exercise 5 - Create the schema on the data warehouse

- Start instance of postgres (from terminal)

```
theia@theiadocker-craigtrupp8:/home/project$ start_postgres
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
37aaf24cf781: Pull complete
Digest:
sha256:9b8dec3bf938bc80fbe758d856e96fdfab5f56c39d44b0cff351e847bb1b01ea
Status: Downloaded newer image for ubuntu:latest
Starting your Postgres database....
This process can take up to a minute.
```

```
Postgres database started, waiting for all services to be ready....

Your Postgres database is now ready to use and available with username:
postgres password: MTE5NTgtY3JhaWd0

You can access your Postgres database via:
 • The Browser with pgadmin
   • URL:
https://craigtrupp8-5050.theiadocker-3-labs-prod-theiak8s-4-tor01.proxy.cog
nitiveclass.ai/browser/
   • Database Password: MTE5NTgtY3JhaWd0
 • CommandLine: psql --username=postgres --host=localhost
```

- Create Database on the Warehouse (Postgres)

```
theia@theiadocker-craigtrupp8:/home/project$ createdb -h localhost -U
postgres -p 5432 billingDW
```

  - -h mentions that the database server is running on the localhost
  - -U mentions that we are using the user name postgres to log into the database
  - -p mentions that the database server is running on port number 5432

- Step 3: Download the schema .sql file.

The commands to create the schema are available in the file below.

```
theia@theiadocker-craigtrupp8:/home/project$ wget
https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB02
60EN-SkillsNetwork/labs/Working%20with%20Facts%20and%20Dimension%20Tables/s
tar-schema.sql
--2023-10-05 11:32:33--
https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB02
60EN-SkillsNetwork/labs/Working%20with%20Facts%20and%20Dimension%20Tables/s
tar-schema.sql
Resolving cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud
(cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud)...
169.63.118.104
Connecting to cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud
(cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud)|169.63.118.104
|:443... connected.
```

```
HTTP request sent, awaiting response... 200 OK
Length: 1105 (1.1K) [application/x-sql]
Saving to: 'star-schema.sql'

star-schema.sql         100%[============================>]   1.08K
--.-KB/s    in 0s

2023-10-05 11:32:33 (154 MB/s) - 'star-schema.sql' saved [1105/1105]
```

- Look at the .sql Schema File

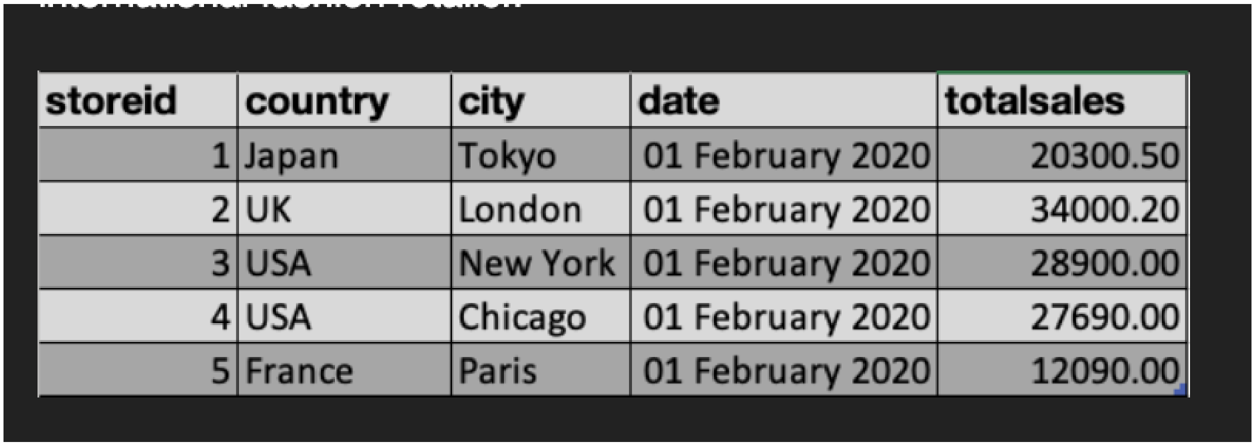
```
BEGIN;


CREATE TABLE public."FactBilling"
(
    rowid integer NOT NULL,
    customerid integer NOT NULL,
    monthid integer NOT NULL,
    billedamount integer NOT NULL,
    PRIMARY KEY (rowid)
);

CREATE TABLE public."DimMonth"
(
    monthid integer NOT NULL,
    year integer NOT NULL,
    month integer NOT NULL,
    monthname "char" NOT NULL,
    quarter integer NOT NULL,
    quartername "char" NOT NULL,
    PRIMARY KEY (monthid)
);

CREATE TABLE public."DimCustomer"
(
    customerid integer NOT NULL,
    category "char" NOT NULL,
    country "char" NOT NULL,
    industry "char" NOT NULL,
    PRIMARY KEY (customerid)
);

ALTER TABLE public."FactBilling"
    ADD FOREIGN KEY (customerid)
    REFERENCES public."DimCustomer" (customerid)
    NOT VALID;


ALTER TABLE public."FactBilling"
    ADD FOREIGN KEY (monthid)
    REFERENCES public."DimMonth" (monthid)
    NOT VALID;

END;
```

- Step 4 : Create the Schema

```
theia@theiadocker-craigtrupp8:/home/project$ psql  -h localhost -U postgres
-p 5432 billingDW < star-schema.sql
BEGIN
CREATE TABLE
CREATE TABLE
CREATE TABLE
ALTER TABLE
ALTER TABLE
COMMIT
```

Practice exercises

In this practice exercise, you will analyze the below csv file, which contains data about the daily sales at different stores of an international fashion retailer.



storeid	country	city	date	totalsales
1	Japan	Tokyo	01 February 2020	20300.50
2	UK	London	01 February 2020	34000.20
3	USA	New York	01 February 2020	28900.00
4	USA	Chicago	01 February 2020	27690.00
5	France	Paris	01 February 2020	12090.00

1. Design the schema for the dimension table **DimStore**.

Field_Name	Details
storeId - (int)	Primary Key - Unique ID for every store
Country - (varChar)	Country where the store is located
City - (varChar)	City where the store is located

2. Design the schema for the dimension table **DimDate**.
    a. Here the customer needs reports to the granularity of a day.
    b. Make sure that you include the day, weekday and weekdayname also in this table.

Field_Name	Details
dateId - (Int)	Primary Key - Id of the date
Day - (Int)	Day from the date field field of the original date
Weekday - (Int)	Weekday from the date field Example: 1, 2, 3, 4, 5, 6, 7. 1 for sunday, 7 for saturday
WeekDayName - Varchar	Weekday name derived from the date field of the original data. Example: Sunday, Monday
Year - int	Year derived from the date field of the original data. Example: 2010
Month - Int	Month number derived from the date field of the original data. Example: 1, 2, 3
MonthName - Varchar	Month name derived from the date field of the original data. Example: March

Quarter - Int	Quarter number derived from the date field of the original data. Example: 1, 2, 3, 4
QuarterName - Varchar	Quarter name derived from the date field of the original data. Example: Q1, Q2, Q3, Q4

- Design the schema for the fact table **FactSales**.
    - Make sure that the **totalsales** field is captured and there is a way to refer to the **store** and the **date**.
    - Also add a rowid to uniquely identify every row.

Field_Name	Details
saleId - Int	Primary Key - Unique row for each individual sale
totalSales - Decimal(6,2)	Values for a store's total sale value for the Day from the CSV file
storeId - Int	Foreign Key - Unique identifier from **DimStore** table defined above ● DimStore.storeId
dateId - Int	Foreign Key - Unique identifire from **DimDate** table defined above ● DimDate.dateId

- Just for Me - Little Testing w/CommandPrompt
    - https://www.commandprompt.com/education/postgresql-list-all-columns-of-a-specific-table/

218
Answered on Nov 8, 2017
You can use PostgreSQL's interactive terminal Psql to show tables in PostgreSQL.

2. Show tables

Now in Psql you could run commands such as:

- \? list all the commands
- \l list databases
- \conninfo display information about current connection

- \c [DBNAME] connect to new database, e.g., \c template1
- \dt list tables of the public schema
- \dt <schema-name>.* list tables of certain schema, e.g., \dt public.*
- \dt *.* list tables of all schemas
- Then you can run SQL statements, e.g., SELECT * FROM my_table;(Note: a statement must be terminated with semicolon ;)
- \q quit psql

```
theia@theiadocker-craigtrupp8:/home/project$ psql --username=postgres
--host=localhost
psql (15.2 (Ubuntu 15.2-1.pgdg18.04+1), server 13.2)
Type "help" for help.

postgres=# \d
Did not find any relations.
postgres=# \l
postgres=# \d
Did not find any relations.
postgres=# \l
postgres=# USE billingDW;
ERROR:  syntax error at or near "USE"
LINE 1: USE billingDW;
        ^
postgres=# \dt billingDW.*
Did not find any relation named "billingDW.*".
postgres=# \d
Did not find any relations.
postgres=# \;
postgres-# \l
postgres-# \c billingDW
psql (15.2 (Ubuntu 15.2-1.pgdg18.04+1), server 13.2)
You are now connected to database "billingDW" as user "postgres".
billingDW-# \d
            List of relations
 Schema |    Name     | Type  |  Owner
--------+-------------+-------+----------
 public | DimCustomer | table | postgres
 public | DimMonth    | table | postgres
 public | FactBilling | table | postgres
(3 rows)

billingDW-# \d public.DimCustomer
```

```
Did not find any relation named "public.DimCustomer".
billingDW-# \d public.DimCustomer;
Did not find any relation named "public.DimCustomer".
billingDW-# \dt
            List of relations
 Schema |    Name     | Type  |  Owner
--------+-------------+-------+----------
 public | DimCustomer | table | postgres
 public | DimMonth    | table | postgres
 public | FactBilling | table | postgres
(3 rows)

billingDW-# \dt public.*
            List of relations
 Schema |    Name     | Type  |  Owner
--------+-------------+-------+----------
 public | DimCustomer | table | postgres
 public | DimMonth    | table | postgres
 public | FactBilling | table | postgres
(3 rows)

billingDW-# \d FactBilling
Did not find any relation named "FactBilling".
billingDW-# SELECT column_name, data_type
billingDW-# FROM information_schema.columns
billingDW-# WHERE table_schema = 'public' AND table_name = 'DimMonth';
 column_name | data_type
-------------+-----------
 monthid     | integer
 year        | integer
 month       | integer
 monthname   | "char"
 quarter     | integer
 quartername | "char"
(6 rows)

billingDW=# \q
theia@theiadocker-craigtrupp8:/home/project$
```