Test for DWH / BI Analyst
Arooba Jamil Khokhar
Submitted on 6/21/2020

Section A. Describe Yourself

Evaluate yourself on the scale 1 (basic knowledge) to 5 (excellent knowledge). You can provide additional information. Fill other areas into empty lines, for example other languages.

AREA	Knowledge
SQL: stands for structure query language. It is used for communicate with database. It stores the data and later we can access and manipulates the data. It is only language that only communicate with Relational database	4
ETL tools – list at most three 1. Data Connectivity 2. Performance 3. Data Quality	4
Reporting / Analytical Tools – list at most three 1. Zoho Analytics 2. Looker 3. Smart Sheet	4
What database design principles do you know? Entity Relationship data model: we used for graphical model. Where entity is object and attributes are properties Integrity constraint Set the rules to insert update the data without effecting and protect the data against damage and accident. DDL/DML Language: Data definition language: used for ‘DROP’, ‘ALTER’, ‘CREATE’ ‘TRUNCATE’ etc Data Manipulation language use to insert, delete, update, select	4

5) How large was the largest database that you worked with as developer / analyst (on the IT / delivery
side), or user (on the business / client side)? Specify the number of records / dimensions.

I have worked on many large datasets in universities and industries. I used 10,0000 row/dimension

I'm most familiar with Microsoft SQL Server. I started worked for 2 years. I've also supported SQL Server environments, as well as MongoDB.

6) What are the most valuable books / courses have you completed in the last 3 years related to
data warehouses and business intelligence?

Recently, I have completed certification of sql and advanced from course era online course.I already learned sql and database from our universities with practical in different industries.

Preparation Steps

- Install postgresql
- Install pdamin4
- Connect to server
- Create database and schema

1st Step

Find the relationship between 3 tables
Make the schema of tables
Find primary keys
Find foreign keys
Find a link between these table
Set one to one relation
Set one to many relation
Set many to many relation between tables

2nd Step

- Create table
- Set datatypes
- For name use varchar
- For date use date
- For account_id use integer
- Set unique key as primary key
- Set reference key as foreign key

3rd Step

- Import csv file from local computer
- Set file variable same as csv file
- Copy table from local csv path

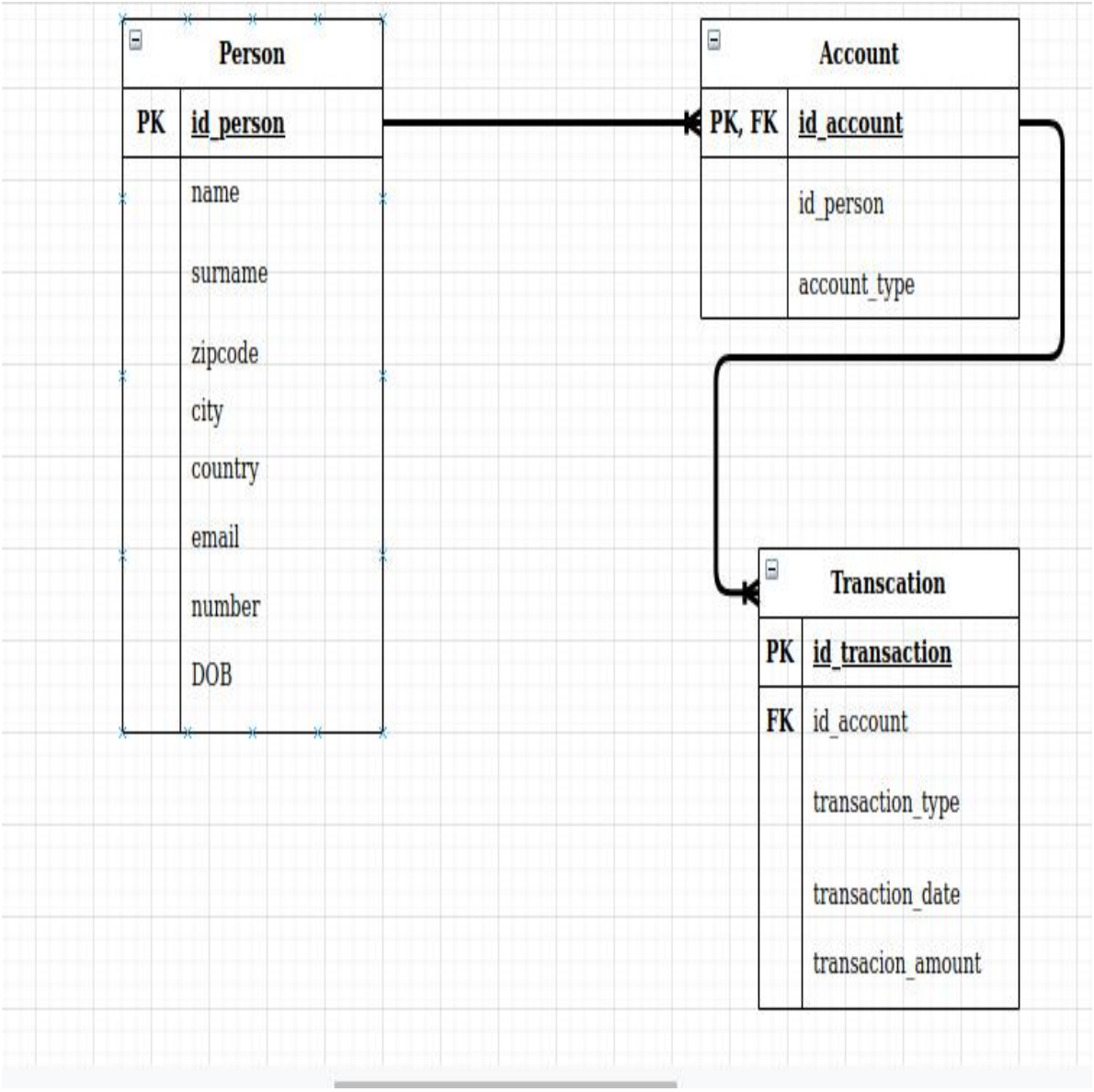
4th Step

- Provide a query that returns transactions for the users 345 and 1234, aggregated monthly, sorted by month, for the period from 15.02.2020 till 06.06.2020:

- select person table
- used left join to join the id_person from account table
- set on a.id_person = p.id_person
- set the clause where id_person should b only(345, 1234)
- select the month and year and remove the date from transaction_date
- using (date_part('month', t.transaction_date) || '.' || date_part('year', t.transaction_date)) as month
- join from d left join transaction as t
- set on d.id_account = t.id_account
- sum the total of transaction round(sum(t.transaction_amount))
- set the range between where t.transaction_date between '2020-02-15' AND '2020-06-06'
- group the month using group by d.id_person, month
- sorted the person_id by order_by

1st step

Entity Relation



2nd Step

Model the data, so that all information can be stored in a relational database. The choice of data types, indices and relations is upon your decision

Person table

```
CREATE TABLE public.person
(
    id_person integer NOT NULL,
    name varchar,
    surname varchar,
    zip integer,
    city character varying COLLATE pg_catalog."default",
    country character varying COLLATE pg_catalog."default",
    email character varying COLLATE pg_catalog."default",
    phone_number integer,
```

Data Output Explain Messages Notifications

```
CREATE TABLE

Query returned successfully in 274 msec.
```

Account

Query Editor Query History

```
CREATE TABLE public.account
(
    id_account integer NOT NULL,
    id_person integer,
    account_type varchar,
    CONSTRAINT account_pkey PRIMARY KEY (id_account),
    CONSTRAINT FK_account FOREIGN KEY (id_person)
        REFERENCES Person(id_person)

)
```

Data Output Explain Messages Notifications

```
CREATE TABLE

Query returned successfully in 303 msec.
```

Transaction

abc/postgres@localhost

Query Editor | Query History

```
1
2
3  CREATE TABLE public.transaction
4  (
5      id_transaction integer NOT NULL,
6      id_account integer,
7      transaction_type varchar,
8      transaction_date date ,
9      transaction_amount float
10
11 --    CONSTRAINT transaction_pkey PRIMARY KEY (id_transaction)
```

Data Output | Explain | Messages | Notifications

```
CREATE TABLE

Query returned successfully in 174 msec.
```

3rd Step

SELECT * FROM table

Query Editor | Query History

```
1  SELECT * FROM PERSON
2  --COPY Public.data FROM '/home/arooba/Documents/master/BI_assignment_csvs (1)/BI_assignment_person.csv' DELIMITE
3
4  --COPY person(id_person,name,surname,zip,city,country,email,phone_number,birth_date)
5  --FROM '/home/arooba/Documents/master/BI_assignment_csvs (1)/BI_assignment_person.csv' CSV HEADER
6  --WHERE id_person is not NULL;
```

Data Output | Explain | Messages | Notifications

	id_person [PK] integer	name character varying	surname character varying	zip integer	city character varying	country character varying	email character varying	phon integ
1	1	Añaterve	Ñeco	10001	les Escaldes	Andorra	añaterve.ñeco@bestdo...	
2	2	Añes	Ñiguez	10002	Andorra la Vella	Andorra	añes.ñiguez@bestdoma...	
3	3	Aadil	Açaola	10003	Umm al Qaywayn	United Arab Emirates	aadil.açaola@bestdom...	
4	4	Aali	Añaños	10004	Ras al-Khaimah	United Arab Emirates	aali.añaños@bestdomai...	
5	5	Aaliyah	Añale	10005	Khawr Fakkān	United Arab Emirates	aaliyah.añale@bestdom...	
6	6	Aaltje	Añazco	10006	Dubai	United Arab Emirates	aaltje.añazco@bestdo...	
7	7	Aamar	Añol	10007	Dibba Al Fujairah	United Arab Emirates	aamar.añol@bestdomai...	

SELECT * FROM account

abc/postgres@localhost

Query Editor | Query History

```
1
2  --COPY account(id_account,id_person,account_type)
3  --FROM '/home/arooba/Documents/master/BI_assignment_csvs (1)/BI_assignment_account.csv' CSV HEADER
4  --WHERE id_account is not NULL;
5
6  SELECT * FROM account
```

Data Output | Explain | Messages | Notifications

	id_account [PK] integer	id_person integer	account_type character varying
285	285	7263	Business Premium
286	286	12092	Business Premium
287	287	11729	Business Premium
288	288	14670	Business Premium
289	289	13643	Business Premium
290	290	7970	Business Premium
291	291	592	Business Premium

SELECT * FROM transaction

```
--COPY Public.data FROM '/home/arooba/Documents/master/BI_assignment_csvs (1)/BI_assignment_person.cs
--COPY transaction (id_transaction,id_account,transaction_type,transaction_date,transaction_amount)
--FROM '/home/arooba/Documents/master/BI_assignment_csvs (1)/BI_assignment_transaction.csv' CSV HEADE
--WHERE id_transaction is not NULL;


--COPY transaction
--FROM '/home/arooba/Documents/master/BI_assignment_csvs (1)/BI_assignment_transaction.csv' CSV HEADE
```

Output | Explain | Messages | Notifications

	id_transaction integer	id_account integer	transaction_type character varying	transaction_date date	transaction_amount double precision
	60	60	P1	2020-03-19	35.3738
	61	61	P1	2020-03-25	20.2144
	62	62	P1	2020-03-27	17.4498
	63	63	P1	2020-03-30	40.2767
	64	64	P1	2020-03-31	33.5437
	65	65	P1	2020-04-07	14.504
	66	66	P1	2020-04-09	35.5534
	67	67	P1	2020-04-15	33.3738

It has no unique id. I did not use any primary key in transaction table. Because it is getting repetition from id_account

0	345	P1	2/21/2020	28.7201
1	345	P1	2/24/2020	33.8123
2	345	P1	3/2/2020	15.4231
3	345	P1	3/6/2020	32.9086
4	345	P1	3/9/2020	24.7128
5	345	P1	3/11/2020	20.6529
6	345	P1	3/19/2020	23.0963
7	345	P1	3/25/2020	7.7015
8	345	P1	3/27/2020	21.7532
9	345	P1	3/30/2020	30.8163
0	345	P1	3/31/2020	2.644
1	345	P1	4/7/2020	1.7953
2	345	P1	4/9/2020	3.9126
3	345	P1	4/15/2020	34.2387
4	345	P1	1/30/2020	8.3228
5	345	P1	2/4/2020	11.356
6	345	P1	2/5/2020	29.6416
7	345	P1	2/17/2020	20.822
8	345	P1	2/18/2020	9.7152
9	345	P1	2/21/2020	30.7848
0	345	P1	2/24/2020	24.4272
1	345	P1	3/2/2020	6.8592

4th step

Provide a query that returns transactions for the users 345 and 1234, aggregated monthly, sorted by month, for the period from 15.02.2020 till 06.06.2020:

id_person [PK] integer	month text	sum_of_transactions double precision
1234	2.2020	553.2275
1234	3.2020	1223.5367999999999
1234	4.2020	802.5818999999999
1234	5.2020	400.30590000000007
345	2.2020	2644.1411999999996
345	3.2020	6663.237200000001
345	4.2020	3242.386599999999
345	5.2020	1712.435200000001

Query Editor Query History

```
1  WITH d as (
2      select p.id_person, p.name, p.surname, a.account_type, a.id_account
3      from person as p left join account as a
4      on a.id_person = p.id_person
5      where p.id_person in (345, 1234)
6  )
7  select d.id_person, (date_part('month', t.transaction_date) || '.' || date_part('year', t.transaction_date)) as month,
8  round(sum(t.transaction_amount))
9  from d left join transaction as t
10 on d.id_account = t.id_account
11 where t.transaction_date between '2020-02-15' AND '2020-06-06'
12 group by d.id_person, month
13 order by d.id_person desc;t
14
```

There is not transaction_date start from 06.2020 in given data

345	P1	4/30/2020	31.11
345	P1	4/30/2020	12.8981
345	P1	4/30/2020	29.5922
345	P1	5/6/2020	30.5032
345	P1	5/6/2020	38.0914
345	P1	5/6/2020	10.8139
345	P1	5/6/2020	28.3576
345	P1	5/6/2020	25.2411
345	P1	5/6/2020	0.0631
345	P1	5/13/2020	26.2176
345	P1	5/13/2020	22.8948
345	P1	5/13/2020	8.6788
345	P1	5/13/2020	36.7379
345	P1	5/13/2020	6.7419
345	P1	5/13/2020	12.3131
345	P1	5/28/2020	11.1594
345	P1	5/28/2020	22.589
345	P1	5/28/2020	29.8261
345	P1	5/28/2020	4.7006
345	P1	5/28/2020	18.4417
345	P1	5/28/2020	7.1181