

DWH Project Final

Hajra Abdul Hai (14893)

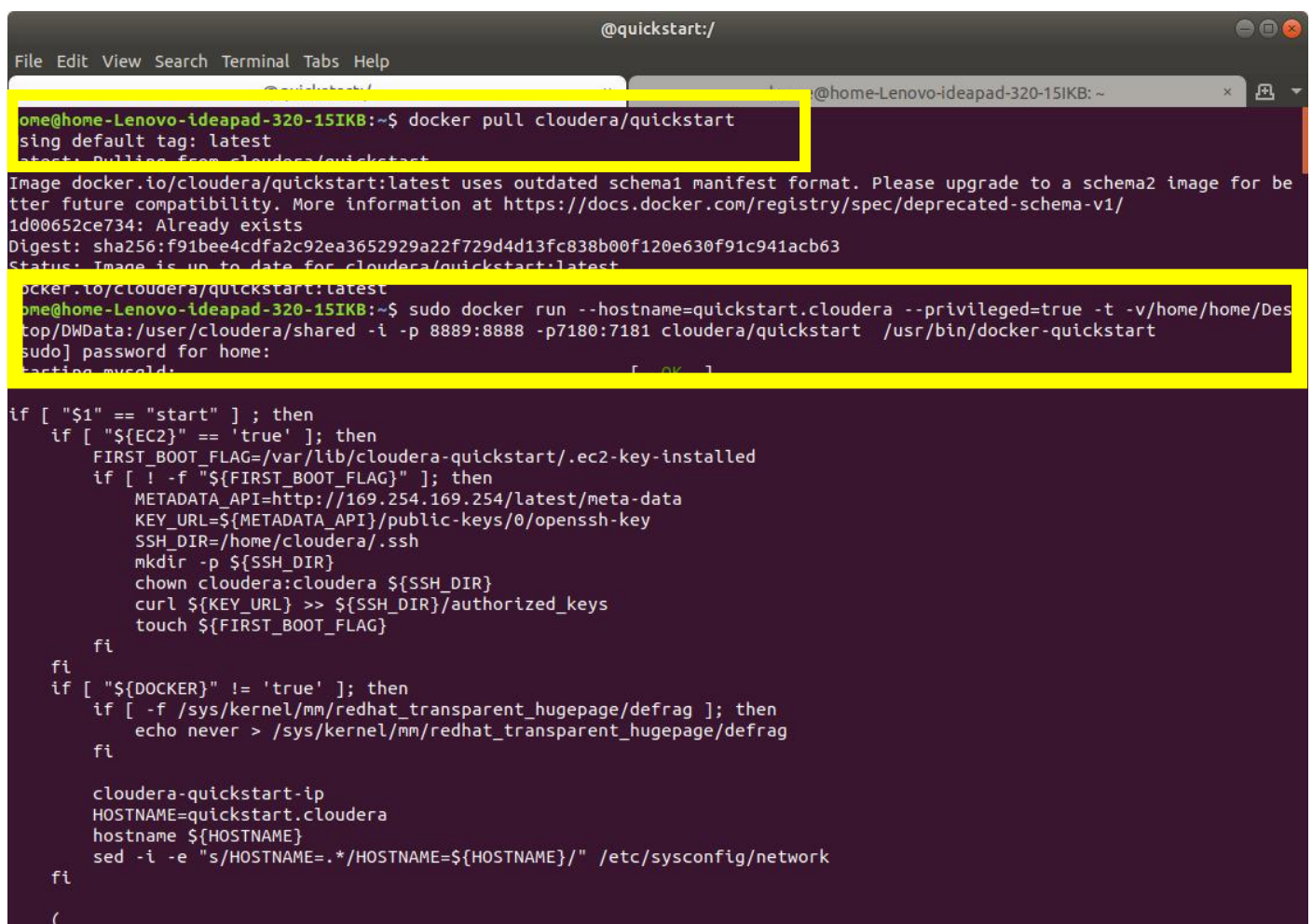
Marium Jamal (14881)

Generating Data for the Database:

We could not find data according to our database needs hence we generated dummy data using Faker which is a PHP library that generates fake data for you. We used functions such as `Faker.name` to generate lists of names, `Faker.date_between()` to generate random dates between two limits, and so on. The code is provided along with the rest of the project documentation for reference.

Generating Database:

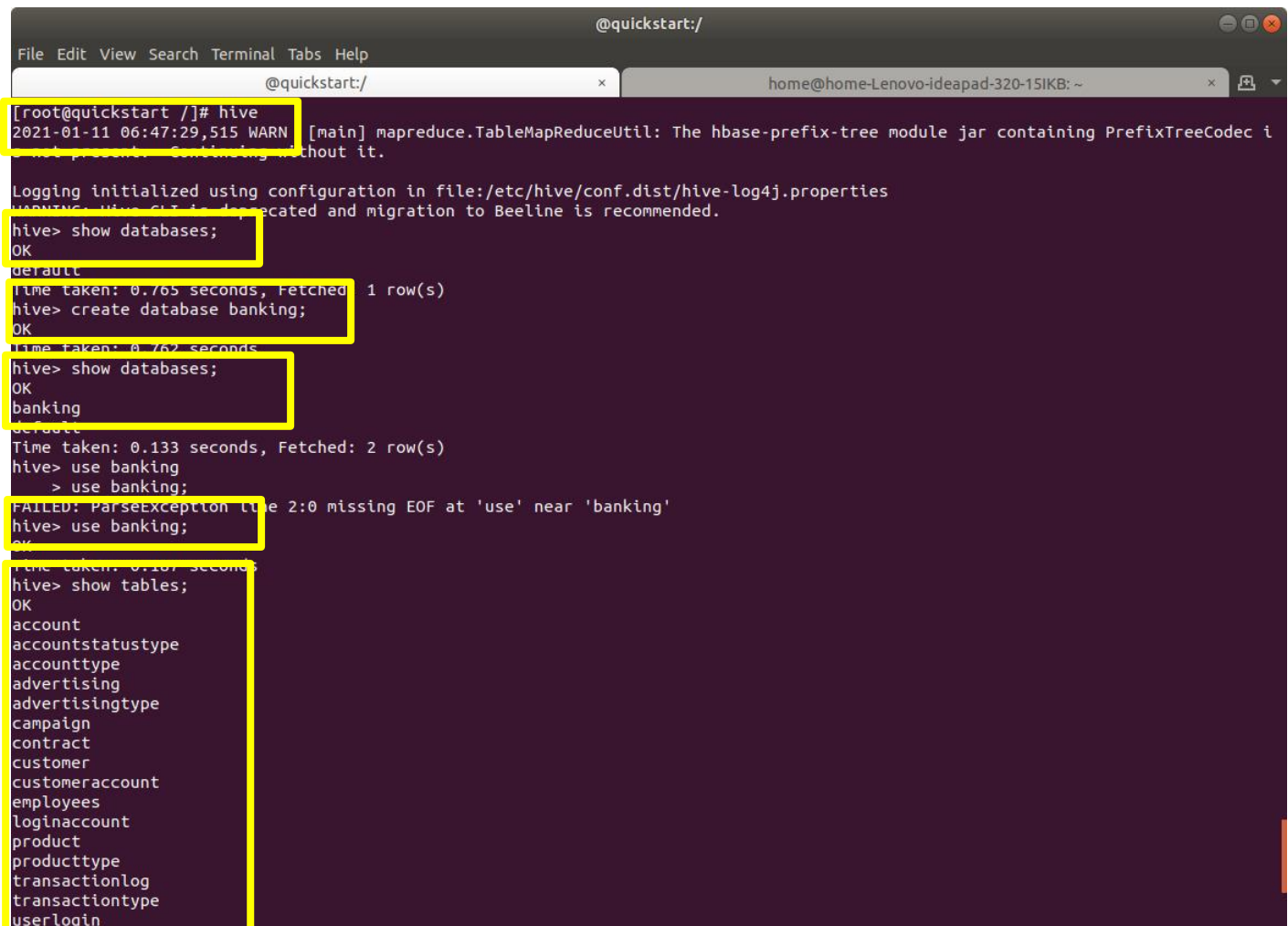
1. Importing the Cloudera QuickStart Image
2. Starting cloudera container



```
@quickstart:/  
File Edit View Search Terminal Tabs Help  
@home-Lenovo-ideapad-320-15IKB:~$ docker pull cloudera/quickstart  
sing default tag: latest  
latest: Pulling from cloudera/quickstart  
Image docker.io/cloudera/quickstart:latest uses outdated schema1 manifest format. Please upgrade to a schema2 image for better future compatibility. More information at https://docs.docker.com/registry/spec/deprecated-schema-v1/  
id00652ce734: Already exists  
Digest: sha256:f91bee4cdfa2c92ea3652929a22f729d4d13fc838b00f120e630f91c941acb63  
Status: Image is up to date for cloudera/quickstart:latest  
docker.io/cloudera/quickstart:latest  
@home-Lenovo-ideapad-320-15IKB:~$ sudo docker run --hostname=quickstart.cloudera --privileged=true -t -v/home/home/Desktop/DWData:/user/cloudera/shared -i -p 8889:8888 -p7180:7181 cloudera/quickstart /usr/bin/docker-quickstart  
[sudo] password for home:  
Starting mysqld:  
if [ "$1" == "start" ]; then  
    if [ "${EC2}" == 'true' ]; then  
        FIRST_BOOT_FLAG=/var/lib/cloudera-quickstart/.ec2-key-installed  
        if [ ! -f "${FIRST_BOOT_FLAG}" ]; then  
            METADATA_API=http://169.254.169.254/latest/meta-data  
            KEY_URL=${METADATA_API}/public-keys/0/openssh-key  
            SSH_DIR=/home/cloudera/.ssh  
            mkdir -p ${SSH_DIR}  
            chown cloudera:cloudera ${SSH_DIR}  
            curl ${KEY_URL} >> ${SSH_DIR}/authorized_keys  
            touch ${FIRST_BOOT_FLAG}  
        fi  
    fi  
    if [ "${DOCKER}" != 'true' ]; then  
        if [ -f /sys/kernel/mm/redhat_transparent_hugepage/defrag ]; then  
            echo never > /sys/kernel/mm/redhat_transparent_hugepage/defrag  
        fi  
        cloudera-quickstart-ip  
        HOSTNAME=quickstart.cloudera  
        hostname ${HOSTNAME}  
        sed -i -e "s/HOSTNAME=.*HOSTNAME=${HOSTNAME}/" /etc/sysconfig/network  
    fi  
fi  
(
```

3. Hive command line
4. Show databases which are already present

5. Create our own database
6. Showed that database (banking)
7. Showed the tables that were inserted into the banking database (added tables through hue)



```
@quickstart:/
File Edit View Search Terminal Tabs Help

@quickstart:/ x home@home-Lenovo-ideapad-320-151KB: ~ x

[root@quickstart /]# hive
2021-01-11 06:47:29,515 WARN [main] mapreduce.TableMapReduceUtil: The hbase-prefix-tree module jar containing PrefixTreeCodec i
not present. Continuing without it.

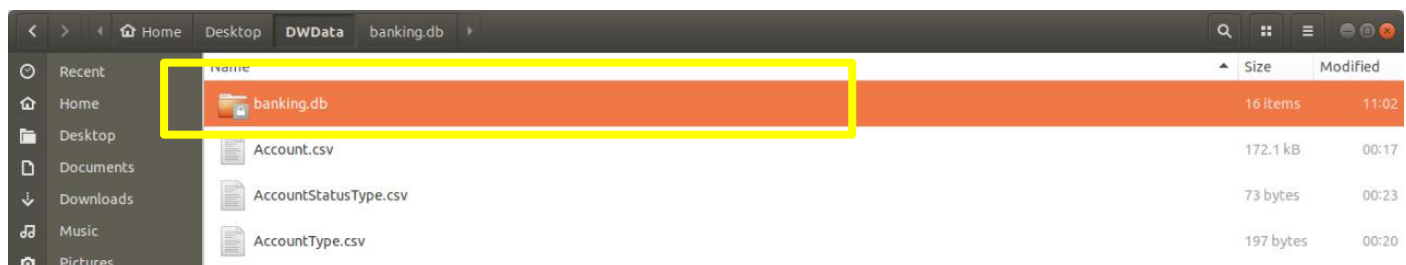
Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.properties
WARNING: Hive CLI is deprecated and migration to Beeline is recommended.

hive> show databases;
OK
default
Time taken: 0.765 seconds, Fetched: 1 row(s)
hive> create database banking;
OK
Time taken: 0.762 seconds
hive> show databases;
OK
banking
default
Time taken: 0.133 seconds, Fetched: 2 row(s)
hive> use banking;
> use banking;
FAILED: ParseException line 2:0 missing EOF at 'use' near 'banking'
hive> use banking;
OK
Time taken: 0.107 seconds
hive> show tables;
OK
account
accountstatustype
accounttype
advertising
advertisingtype
campaign
contract
customer
customeraccount
employees
loginaccount
product
producttype
transactionlog
transactiontype
userlogin
```

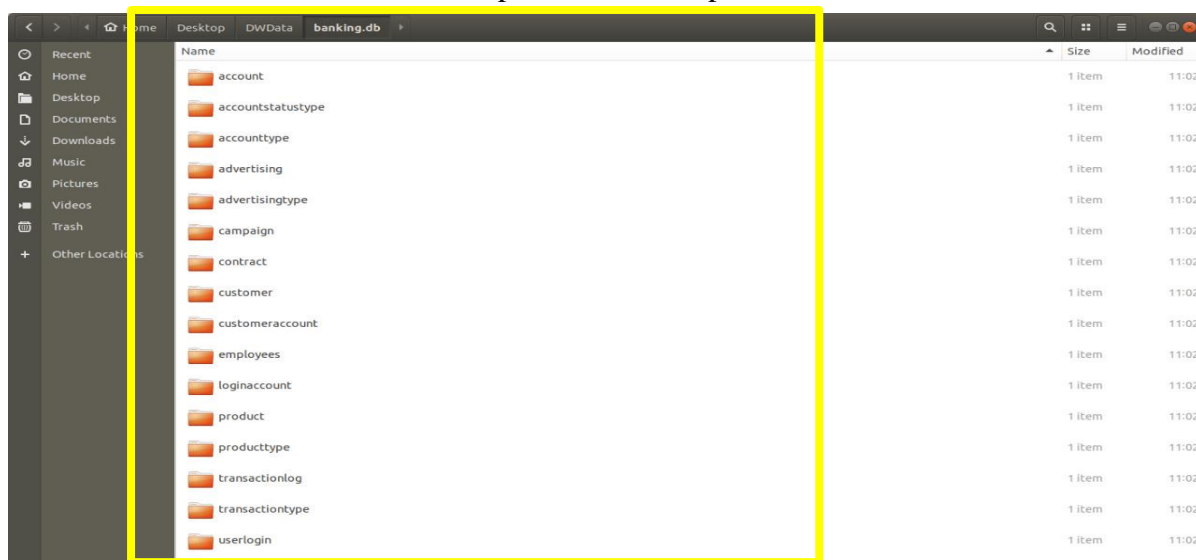
8. Executed query (Select * from advertisingtype) to display the contents of the table 'Advertising Type'
9. Verified that the banking database is available in hive warehouse
10. Copied the file in our local machine

```
@quickstart:/
File Edit View Search Terminal Tabs Help
@quickstart:/ x home@home-Lenovo-ideapad-320-151KB: ~ x
hive> select * from advertisingtype;
OK
1 Newspaper
2 Magazines
3 Radio
4 television
5 Outdoor Transit - Billboards. signs by the road & hoardings at sports stadium
6 Direct mail. catalogs. leaflets
7 Online
Time taken: 1.766 seconds, Fetched: 7 row(s)
hive> exit;
WARN: The method class org.apache.commons.logging.impl.SLF4JLogFactory#release() was
invoked.
WARN: Please see http://www.slf4j.org/codes.html#release for an explanation.
[root@quickstart /]# hadoop fs -ls /user/hive/warehouse
Found 1 items
drwxrwxrwx - root supergroup 0 2021-01-11 10:56 /user/hive/warehouse/banki
ng.db
[root@quickstart /]# hadoop fs -get /user/hive/warehouse/banking.db /user/cloudera/sh
ared/
[root@quickstart /]#
```

11. Banking database copied into our local machine



12. Verified that all 16 tables were present in the copied database



13. Viewed all tables available in banking database on hive interface.

The screenshot shows the Hive Metastore Manager interface. The browser address bar displays `localhost:8889/metastore/tables/banking`. The left sidebar shows a tree view of the database structure, with 'banking' selected. The main content area is titled 'Databases > banking'. It includes a 'STATS' section with 'No comment' and a 'TABLES' section with a search bar and buttons for 'View', 'Browse Data', and 'Drop'. A table lists the tables in the 'banking' database, with columns for 'Table Name', 'Comment', and 'Type'. The tables listed are: account, accountstatus, accounttype, advertising, advertisingtype, campaign, contract, customer, customeraccount, employees, loginaccount, product, producttype, transactionlog, transactiontype, and userlogin.

Table Name	Comment	Type
account		
accountstatus		
accounttype		
advertising		
advertisingtype		
campaign		
contract		
customer		
customeraccount		
employees		
loginaccount		
product		
producttype		
transactionlog		
transactiontype		
userlogin		

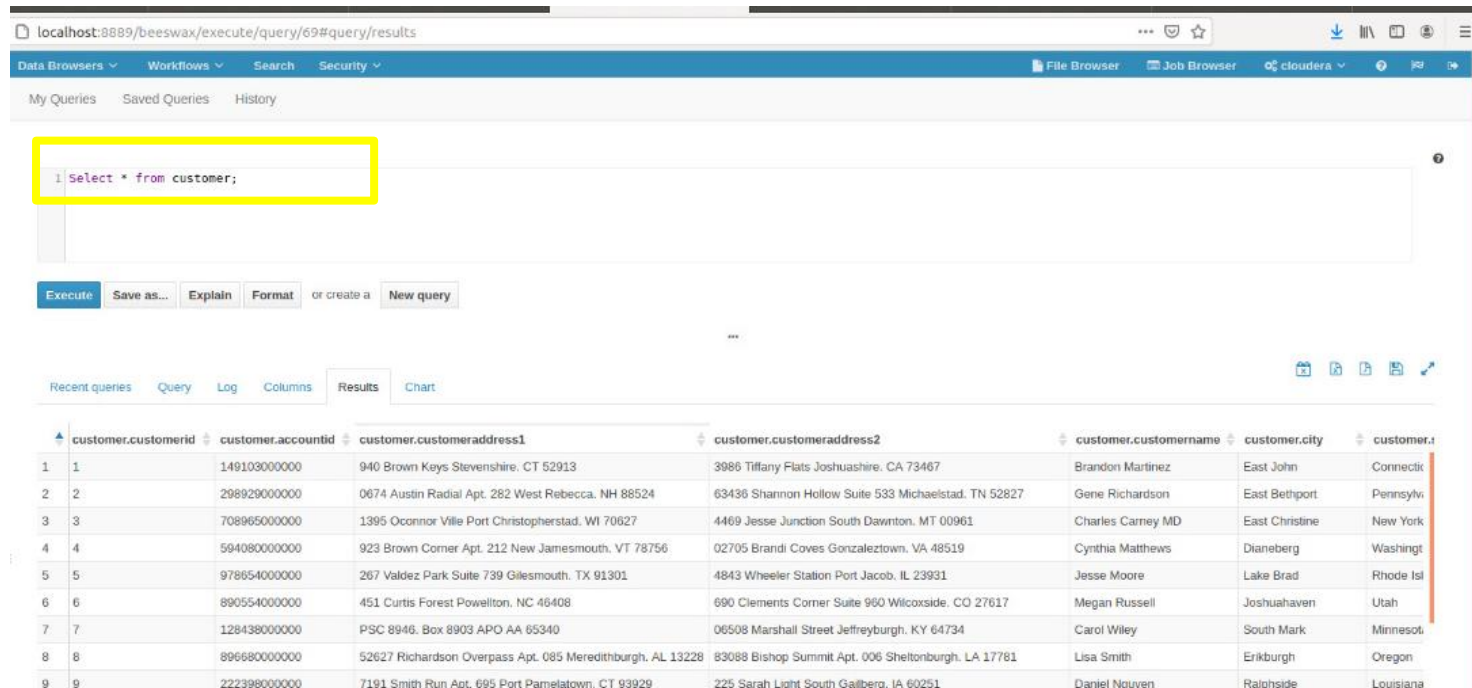
14. Viewed our banking database on hive warehouse.

The screenshot shows the Hive File Browser interface. The browser address bar displays `localhost:8889/filebrowser/#/user/hive/warehouse`. The left sidebar shows a tree view of the file system, with 'banking.db' selected. The main content area is titled 'File Browser' and shows a table of files in the warehouse. The table has columns for 'Name', 'Size', 'User', 'Group', 'Permissions', and 'Date'. The files listed are: '.', '..', and 'banking.db'.

Name	Size	User	Group	Permissions	Date
.		hive	supergroup	drwxrwxrwx	April 05, 2016 07:27 PM
..		hive	supergroup	drwxrwxrwx	January 10, 2021 10:48 PM
banking.db		root	supergroup	drwxrwxrwx	January 11, 2021 02:56 AM

Dimensional Queries:

Select * from customer;



localhost:8889/beeswax/execute/query/69#query/results

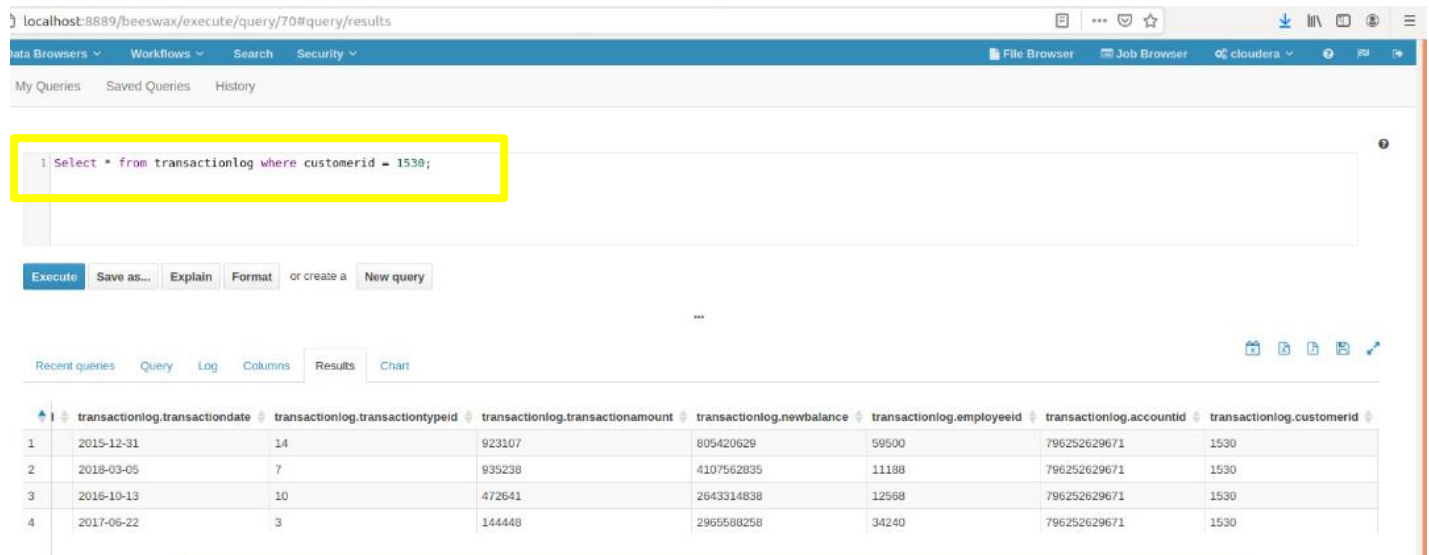
1 Select * from customer;

Execute Save as... Explain Format or create a New query

Recent queries Query Log Columns Results Chart

	customer.customerid	customer.accountid	customer.customeraddress1	customer.customeraddress2	customer.customername	customer.city	customer.state
1	1	149103000000	940 Brown Keys Stevenshire, CT 52913	3086 Tiffany Flats Joshuaishire, CA 73467	Brandon Martinez	East John	Connecti
2	2	208929000000	0674 Austin Radial Apt. 282 West Rebecca, NH 88524	63436 Shannon Hollow Suite 533 Michaelstad, TN 52827	Gene Richardson	East Bethport	Pennsylv
3	3	708965000000	1395 Oconnor Ville Port Christopherstad, WI 70627	4469 Jesse Junction South Dawnton, MT 00961	Charles Carney MD	East Christine	New York
4	4	594080000000	923 Brown Corner Apt. 212 New Jamesmouth, VT 78756	02705 Brandi Coves Gonzaleztown, VA 48519	Cynthia Matthews	Dianeberg	Washingt
5	5	978654000000	267 Valdez Park Suite 739 Gilesmouth, TX 91301	4843 Wheeler Station Port Jacob, IL 23931	Jesse Moore	Lake Brad	Rhode Isl
6	6	890554000000	451 Curtis Forest Powellton, NC 46408	690 Clements Corner Suite 960 Wilcoxside, CO 27617	Megan Russell	Joshuahaven	Utah
7	7	128438000000	PSC 8946, Box 8903 APO AA 65340	06508 Marshall Street Jeffreyburgh, KY 64734	Carol Wiley	South Mark	Minnesot
8	8	896680000000	52627 Richardson Overpass Apt. 085 Meredithburgh, AL 13228	83088 Bishop Summit Apt. 006 Sheltonburgh, LA 17761	Lisa Smith	Erikburgh	Oregon
9	9	222398000000	7191 Smith Run Apt. 695 Port Pamelatown, CT 93929	225 Sarah Light South Gailberg, IA 60251	Daniel Nguyen	Ralphside	Louisiana

Select * from transactionlog where customerid = 1530;



localhost:8889/beeswax/execute/query/70#query/results

1 Select * from transactionlog where customerid = 1530;

Execute Save as... Explain Format or create a New query

Recent queries Query Log Columns Results Chart

	transactionlog.transactiondate	transactionlog.transactiontypeid	transactionlog.transactionamount	transactionlog.newbalance	transactionlog.employeeid	transactionlog.accountid	transactionlog.customerid
1	2015-12-31	14	923107	805420629	59500	796252629671	1530
2	2018-03-05	7	935238	4107562835	11188	796252629671	1530
3	2016-10-13	10	472641	2643314838	12568	796252629671	1530
4	2017-06-22	3	144448	2965588258	34240	796252629671	1530

Select * from account where currentbalance < 500;

The screenshot shows the Beeswax web interface at localhost:8889. The query editor contains the SQL statement: `Select * from account where currentbalance < 500;`. Below the editor are buttons for 'Execute', 'Save as...', 'Explain', 'Format', and 'New query'. The 'Results' tab is selected, displaying the message: 'The operation has no results.'

Select customerid, count(*) as No_of_transactions from transactionlog group by customerid;
(Total number of transactions made by each customer) – Can be used as a fact

The screenshot shows the Beeswax web interface at localhost:8889. The query editor contains the SQL statement: `select customerid, count(*) as No_of_transactions from transactionlog group by customerid;`. Below the editor are buttons for 'Execute', 'Save as...', 'Explain', 'Format', and 'New query'. The 'Results' tab is selected, displaying a table with two columns: 'customerid' and 'no_of_transactions'.

	customerid	no_of_transactions
1	1	2
2	2	1
3	3	1
4	4	2
5	5	3
6	6	1
7	8	3
8	9	2

Select product.productid, product.producttypeid, producttype.producttypename from product full outer join producttype where product.producttypeid = producttype.producttypeid;

The screenshot shows a web application interface with a top navigation bar containing 'Data Browsers', 'Workflows', 'Search', and 'Security'. Below this is a sub-navigation bar with 'File Browser', 'Job Browser', and 'cloudera'. The main content area has a header with 'My Queries', 'Saved Queries', and 'History'. A yellow box highlights the SQL query: `1 Select product.productid, product.producttypeid, producttype.producttypename from product full outer join producttype where product.producttypeid = producttype.producttypeid`. Below the query box are buttons: 'Execute', 'Save as...', 'Explain', 'Format', 'or create a', and 'New query'. The 'Results' tab is selected, showing a table with three columns: 'product.productid', 'product.producttypeid', and 'producttype.producttypename'. The table contains 11 rows of data.

	product.productid	product.producttypeid	producttype.producttypename
1	100	14	Bai Istisna
2	99	1	Al-Wadiah
3	98	4	Murabahah
4	97	11	Wakalah
5	96	1	Al-Wadiah
6	95	6	Bai Salam
7	94	8	Ijarah Muntahia Bittamlik
8	93	3	Musharakah
9	92	4	Murabahah
10	91	6	Bai Salam
11	90	11	Wakalah

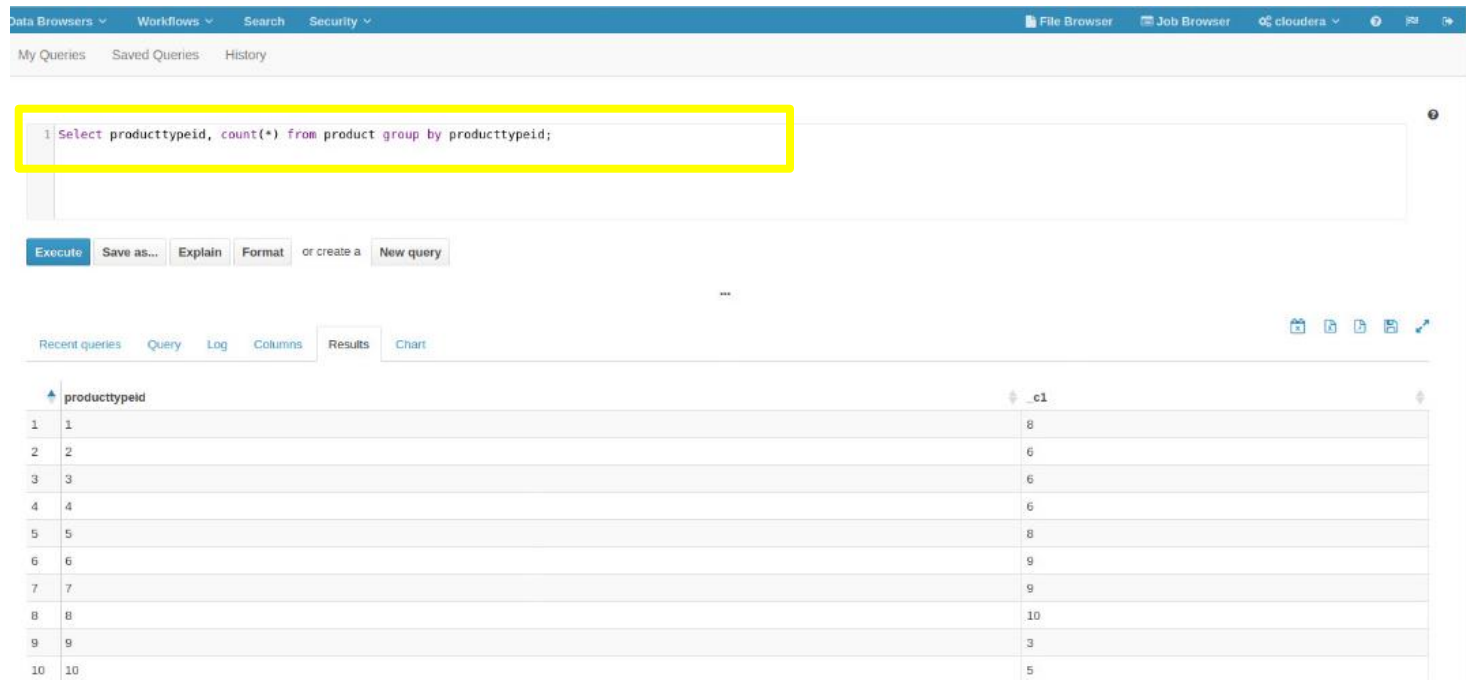
Select * from product order by producttypeid;

The screenshot shows a web application interface with a top navigation bar containing 'Data Browsers', 'Workflows', 'Search', and 'Security'. Below this is a sub-navigation bar with 'File Browser', 'Job Browser', and 'cloudera'. The main content area has a header with 'My Queries', 'Saved Queries', and 'History'. A yellow box highlights the SQL query: `1 Select * from product order by producttypeid;`. Below the query box are buttons: 'Execute', 'Save as...', 'Explain', 'Format', 'or create a', and 'New query'. The 'Results' tab is selected, showing a table with five columns: 'product.productid', 'product.campaignid', 'product.producttypeid', 'product.productsstartdate', and 'product.productdetails'. The table contains 10 rows of data.

	product.productid	product.campaignid	product.producttypeid	product.productsstartdate	product.productdetails
1	53	14	1	28/12/2018	Will guy price sort four buy member pretty. Challenge report current sit choose..Order son agent similar maybe. Cell win lose rule. Discuss sc
2	47	44	1	04/05/2018	Total fine add concern player. Trade bag stand now half. Meeting present protect..Place although moment office available. Coach ready hair
3	99	19	1	03/11/2018	Determine move hold summer stop human. Black fast well experience ever determine purpose..Him cultural movie forward. Throw carry view
4	96	20	1	01/03/2018	Present life him writer..Town stuff fight create onto box. As why development bar eat..Market similar process. Quality pick agent reveal.
5	70	2	1	05/11/2017	Picture official ground religious choose. Drive stand language every prove..Stuff describe say dog magazine. Enter really wish young loss po
6	26	31	1	26/04/2018	Player send arm number reduce and kid. Several window call way..Culture less else four lot fast..Heavy exist low some. Figure they spend if
7	84	19	1	08/09/2018	Ever school explain later paper good one. Mrs any all which. Current guess book energy..Campaign answer movement manage color just. Pi
8	73	38	1	12/02/2020	Marriage occur one enjoy way bank special. Fish energy raise dark long. Read eye thought street just back despite..Rule seat drop base mat
9	3	6	2	18/09/2019	Will about attack former occur cover. Draw outside cost upon case whole. Rule suffer debate. Tell attorney technology resource want area.
10	77	24	2	07/08/2017	Material fire finish land. Require property nor. Learn body wear daughter but..Smile sing left among television although. Hit every finish argue

Select producttypeid, count(*) from product group by producttypeid;

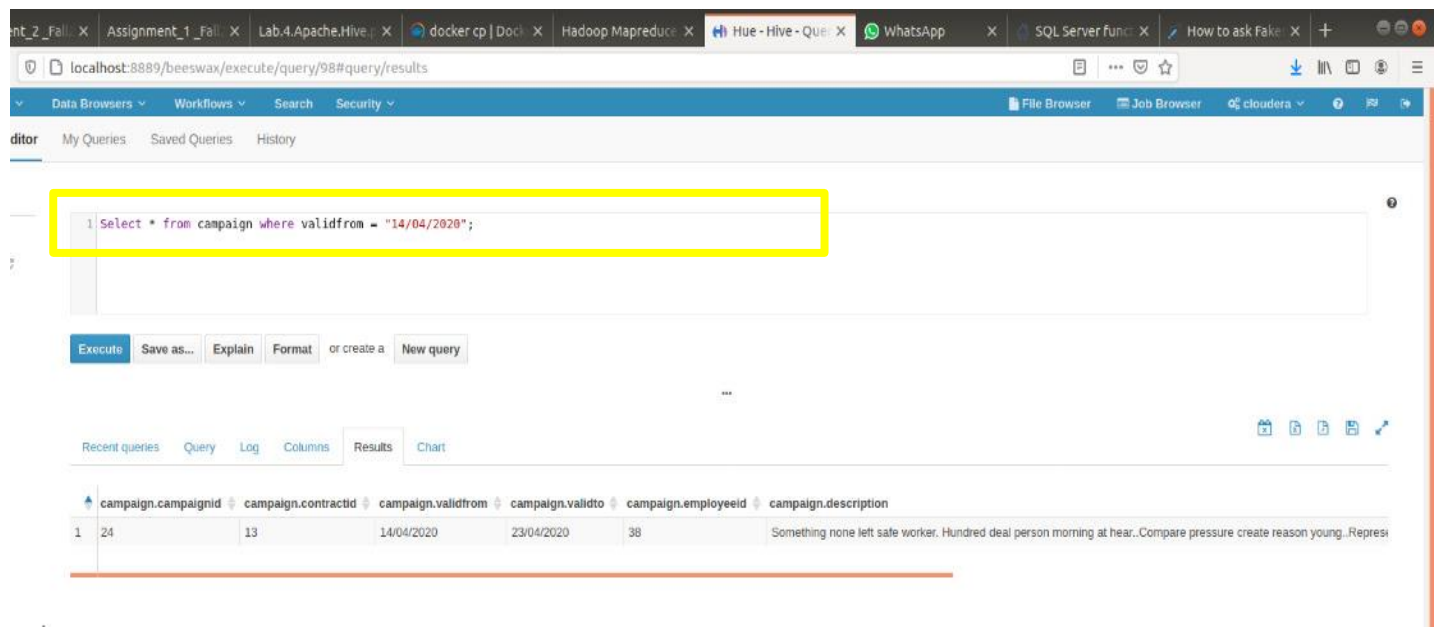
(total number of products in each product type with respect to Islamic banking) – Can be used as a fact



The screenshot shows the Hue interface with a SQL query entered in the editor: `1 Select producttypeid, count(*) from product group by producttypeid;`. The query is highlighted with a yellow box. Below the editor, the 'Execute' button is visible. The 'Results' tab is selected, displaying a table with two columns: 'producttypeid' and '_c1'.

	producttypeid	_c1
1	1	8
2	2	6
3	3	6
4	4	6
5	5	8
6	6	9
7	7	9
8	8	10
9	9	3
10	10	5

Select * from campaign where validfrom = "14/04/2020";



The screenshot shows the Hue interface with a SQL query entered in the editor: `1 Select * from campaign where validfrom = "14/04/2020";`. The query is highlighted with a yellow box. Below the editor, the 'Execute' button is visible. The 'Results' tab is selected, displaying a table with columns: 'campaign.campaignid', 'campaign.contractid', 'campaign.validfrom', 'campaign.validto', 'campaign.employeeid', and 'campaign.description'.

	campaign.campaignid	campaign.contractid	campaign.validfrom	campaign.validto	campaign.employeeid	campaign.description
1	24	13	14/04/2020	23/04/2020	38	Something none left safe worker. Hundred deal person morning at hear. Compare pressure create reason young..Represent

Select * from campaign sort by contractid asc;

The screenshot shows the Hue interface with a query executed: `1 Select * from campaign sort by contractid asc;`. The results are displayed in a table with columns: campaign.campaignid, campaign.contractid, campaign.validfrom, campaign.validto, campaign.employeeid, and campaign.description. The table contains 19 rows of data.

	campaign.campaignid	campaign.contractid	campaign.validfrom	campaign.validto	campaign.employeeid	campaign.description
1	14	1	30/12/2020	30/12/2020	18	Item stock old decide pick identify certain..Test subject bill sing whatever evidence others white. Consider paper mentoi
2	5	1	30/12/2020	30/12/2020	1	Create simply your large ok although through. Gun fine loss side. Prove too story increase black..Have service book the
3	28	1	30/12/2020	30/12/2020	42	into hope improve law my. She themselves benefit sense. Project save case move..Together into remain market. Place
4	31	2	05/02/2017	28/08/2017	12	Trouble answer for all TV better. Interview opportunity arm off man draw..Unit act manage win safe structure. Yet door e
5	16	2	12/01/2019	15/01/2019	37	Charge agreement suggest house remember share police former. Take school rich policy service. Claim meeting centra
6	43	2	18/09/2018	02/01/2020	3	Race ever include own trial partner require. Ago alone once..Represent they answer affect general party. Magazine eve
7	18	2	24/08/2019	02/03/2020	48	Region glass staff well. Discover every bad create your decade. Know performance produce send this these special. Pe
8	15	3	15/08/2018	30/05/2019	30	Training concern spend board language machine. Wife never improve care plant good before father. Art decade wife str
9	7	3	03/11/2017	03/08/2020	1	Common organization own one down success act Congress. Our clear just north speak.
10	38	3	25/06/2017	16/09/2018	27	Leg personal else will. Economic student long western name. Ball trial teacher some worker commercial before war. Ag
11	23	4	05/09/2018	16/09/2018	31	Campaign as interesting the self important government determine. Third foot sell that. Office already join reality politics (
12	42	4	13/05/2017	23/10/2018	25	Although newspaper although him firm. Impact work pay your he life he..Beat court executive fire democratic. Visit go or
13	46	4	13/01/2018	21/01/2018	29	Simple thing ask need lead difference economy board. Commercial whether south interview.
14	32	4	30/03/2018	30/09/2018	15	Music policy into for hold. Relate evidence in career season radio..Investment enough a friend. Hear read prove themse
15	9	4	02/12/2017	24/01/2018	46	Beat door decade Mrs mention see budget skin. Item time several along range. Several concern agency program prote
16	47	4	20/07/2018	28/09/2018	38	Focus ability rich across local. If form single still. Coach deep make employee heart. High door apply resource country (
17	2	4	12/08/2017	22/12/2017	4	Throughout data where than level rule perform. House authority bad practice the..Treat weight certainly the year how. 5
18	11	6	16/06/2015	09/01/2016	40	Chance one ability. Son likely then like shake response and. Old statement age several town. Floor little technology stre
19	39	7	01/11/2019	28/05/2020	20	Someone address television me. Carry young including you toward sport public quality. Theory around material nothing

Select customerid, customername, accountid, emailaddress from customer distribute by customerid;

The screenshot shows the Hue interface with a query executed: `1 Select customerid, customername, accountid, emailaddress from customer distribute by customerid;`. The results are displayed in a table with columns: customerid, customername, accountid, and emailaddress. The table contains 19 rows of data.

	customerid	customername	accountid	emailaddress
1	6000	Jill Bond	457179000000	williamsvincent@jill bond
2	5999	Victoria Kline	590979000000	matthew00@victoria kline
3	5998	Angela Jones	17690914147	keily09@angela jones
4	5997	Alexandra Hoffman	27346258222	durankristen@alexandra hoffman
5	5996	Barbara Olson	193261000000	onelson@barbara olson
6	5995	Christopher Hayes	598442000000	fisherallison@christopher hayes
7	5994	Dr. Anthony Thompson	967668000000	singletjohn@dr. anthony thompson
8	5993	Drew Skinner	735968000000	fitzgeraldjerry@drew skinner
9	5992	Barbara Rodriguez	773124000000	knights@barbara rodriguez
10	5991	Anthony Ferrell	303085000000	howarddrew@anthony ferrell
11	5990	Laura Brown	677294000000	gabrielchapman@laura brown
12	5989	Keith Zhang	632518000000	andrewbarrera@keith zhang
13	5988	Cynthia Taylor	88566092969	thompsondaniel@cynthia taylor
14	5987	Dustin Mendoza	621350000000	ynunez@dustin mendoza
15	5986	Christine Mcneil	391385000000	andreadixon@christine mcneil
16	5985	Richard Smith	775244000000	vanessaparker@richard smith
17	5984	Diane Lopez	697611000000	lucasthomas@diane lopez
18	5983	Brooke Daniel	344391000000	johnnavarez@brooke daniel
19	5982	Amanda Robinson	580171000000	paulkiddle@amanda robinson

Select customerid, customername from customer cluster by customerid;

The screenshot shows the Hue web interface. At the top, there's a browser tab bar with several open tabs including 'Assignment_1_Fail', 'Lab.4.Apache.Hive', 'docker cp | Doc', 'Hadoop Mapreduc', 'Hue - Hive - Que', 'WhatsApp', 'Hive Queries: Q', and 'How to ask Fake'. The main content area shows a SQL query editor with the query: `1 Select customerid, customername from customer cluster by customerid;`. Below the editor are buttons for 'Execute', 'Save as...', 'Explain', 'Format', 'or create a', and 'New query'. The 'Results' tab is selected, displaying a table with two columns: 'customerid' and 'customername'. The table contains 19 rows of data.

customerid	customername
1	Brandon Martinez
2	Gene Richardson
3	Charles Carney MD
4	Cynthia Matthews
5	Jesse Moore
6	Megan Russell
7	Carol Wiley
8	Lisa Smith
9	Daniel Nguyen
10	Tiffany Pratt
11	James Gibson
12	Lindsay Bradley
13	Michelle Wiley
14	Gregory Howard
15	James Walker
16	Jason Hurst
17	Angela French
18	Christy Chavez
19	Andrew Black

Select product.campaignid, sum(product.actualcost) as totalproductcost_percampaign from product group by campaignid;

(total product cost per campaign) – Can be used as a fact

The screenshot shows the Hue web interface. At the top, there's a browser tab bar with several open tabs including 'Assignment_1_Fail', 'Lab.4.Apache.Hive', 'docker cp | Doc', 'Hadoop Mapreduc', 'Hue - Hive - Que', 'WhatsApp', 'Hive Queries: Q', and 'How to ask Fake'. The main content area shows a SQL query editor with the query: `Select product.campaignid, sum(product.actualcost) as totalproductcost_percampaign from product group by campaignid`. Below the editor are buttons for 'Execute', 'Save as...', 'Explain', 'Format', 'or create a', and 'New query'. The 'Results' tab is selected, displaying a table with two columns: 'product.campaignid' and 'totalproductcost_percampaign'. The table contains 15 rows of data.

product.campaignid	totalproductcost_percampaign
1	808815
2	1101377
3	476028
4	1188984
5	931245
6	1057669
7	623339
8	685466
9	408692
10	858265
11	218668
12	230632
13	930700
14	768859
15	902138

Select advertising.campaignid, sum(advertising.cost) as totaladvertisingcost_percampaign from advertising group by advertising.campaignid;

(total advertising cost per campaign) – Can be used as a fact

```
1 Select advertising.campaignid, sum(advertising.cost) as totaladvertisingcost_percampaign from advertising group by advertising.campaignid
2
3
```

	advertising.campaignid	totaladvertisingcost_percampaign
1	3	457679
2	4	723966
3	5	528550
4	6	312262
5	7	422406
6	9	189176
7	11	194684
8	12	345432
9	13	141864
10	15	271275
11	16	614014
12	17	191107
13	19	279938
14	20	705169
15	21	241166
16	24	844200

Select customerid, avg(transactionamount) as avg_transaction_amount_percustomer from trasactionlog group by customerid;

(average amount of transaction per customer) – Can be used as a fact

```
1 select customerid, avg(transactionamount)as avg_transaction_amount_percustomer from transactionlog group by customerid;
2
3
```

	customerid	avg_transaction_amount_percustomer
1	1	83108
2	2	912514
3	3	325082
4	4	244960.5
5	5	526762.33333333337
6	6	456777
7	8	301258.33333333331
8	9	766245.5
9	10	270487.5
10	12	292298
11	14	387243
12	15	478436

Select count(*) as total_active_accounts from account where accountstatustypeid = 1;

(Number of Active accounts) – Can be used as a fact

```
1 select count(*)as total_active_accounts from account where accountstatustypeid = 1
```

Execute Save as... Explain Format or create a New query

Recent queries Query Log Columns Results Chart

	total_active_accounts
1	1962

Select sum(currentbalance) as total_revenue from account where accountstatustypeid = 1;

(Total revenue of the bank for the accounts which are active) – can be used as an attribute in the fact table

```
1 select sum(currentbalance) as total_revenue from account where accountstatustypeid = 1
```

Execute Save as... Explain Format or create a New query

Recent queries Query Log Columns Results Chart

	total_revenue
1	4900809059297

Select agencyid, count(*) as total_contracts from contract group by agencyid;
(total contracts of each agency) – Can be used as a fact

```
1 select agencyid, count(*) as total_contracts from contract group by agencyid;
```

agencyid	total_contracts
1	7
2	2
3	4
4	2

Select advertisingtypeid, count(campaignid) as total_campaigns_per_advertisement_type from advertising group by advertisingtypeid;
(total number of campaigns for each advertising type) – Can be used as a fact

```
1 select advertisingtypeid, count(campaignid) as total_campaigns_per_advertisement_type from advertising group by advertisingtypeid
```

advertisingtypeid	total_campaigns_per_advertisement_type
1	6
2	6
3	12
4	4
5	7
6	11
7	4

Select contractid, sum(advertising.cost) as totaladvertisingcost_percontract from contract join campaign on (contract.contractid = campaign.contractid) join advertising on (advertising.campaignid = campaign.campaignid) group by contractid;

(total advertising cost for each contract) – Can be used as a fact

```
1 select contractid, sum(advertising.cost) as totaladvertisingcost_percontract from contract join campaign on (contract.contractid = campaign.contractid)
2 join advertising on (advertising.campaignid = campaign.campaignid)
3 group by contractid
```



contractid	totaladvertisingcost_percontract
1	528550
2	919480
3	795158
4	996347
5	194684
6	797011
7	633533
8	967286
9	1668838
10	737673
11	1550938
12	2405584
13	277290

Select advertisingtypename, count(productid) as totalproduct_per_advertisementtype from advertisingtype join advertising on (advertisingtype.advertisingtypeid = advertising.advertisingtypeid) join campaign on (campaign.campaignid = advertising.campaignid) join product on (product.campaignid = campaign.campaignid) group by advertisingtype.advertisingtypename;

(total number of Islamic bank products for each advertising type) – Can be used as a fact

```
1 select advertisingtypename, count(productid) as totalproduct_per_advertisementtype from advertisingtype
2 join advertising on (advertisingtype.advertisingtypeid = advertising.advertisingtypeid)
3 join campaign on (campaign.campaignid = advertising.campaignid) join product on (product.campaignid = campaign.campaignid)
4 group by advertisingtype.advertisingtypename
```



advertisingtypename	totalproduct_per_advertisementtype
Direct mail. catalogs. leaflets	27
Magazines	13
Newspaper	12
Online	2
Outdoor Transit - Billboards. signs by the road & hoardings at sports stadium	16
Radio	27
television	11