Graph & Modern Databases

Akeem Babalola

Big Data and Business Intelligence

1. Cassandra

The concept behind this database is an application called ‘Chippi’ which acts as an educational platform for students and teachers, allowing for collaboration between users in participation of industry accredited courses. The columns used in the table are labelled accordingly to describe characteristics related to each user, with 35 rows of user data. The scalability of this structure can be considered a benefit as it allows for the storage of user information and logs that provide insight into the app activity of users. Also, the advanced search feature allows for efficient retrieval of information for users seeking specific aspects. A disadvantage of this design could be the relative complexity which requires the use of partition keys to allow filtering. I have pasted screenshots below to evidence the code and examples of queries ranging in complexity.

```
[nosql@nosql Desktop]$ cqlsh
Connected to Test Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.11.13 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
```

Creation of keyspace (application named chippi) and primary database (userbase). User ID is listed as a primary key as this uniquely identifies each user:

```
cqlsh> CREATE KEYSPACE chippi WITH REPLICATION = {'class': 'SimpleStrategy', 'replication_factor':1};
cqlsh> USE chippi;
cqlsh:chippi> CREATE TABLE userbase (
          ... user_id varchar PRIMARY KEY,
          ... first_name varchar,
          ... last_name varchar,
          ... gender varchar,
          ... account_type varchar,
          ... region varchar,
          ... industry_of_choice varchar,
          ... screen_time time,
          ... credit_balance double,
          ... real_budget_spend double,
          ... premium_account boolean);
cqlsh:chippi>
```

First of data entry showcasing the input values used to populate each field, batch entry will be utilised throughout the creation of this database:

```
cqlsh:chippi> INSERT INTO userbase (user_id, first_name, last_name, gender, account_type, region, industry_of_choice, screen_time, credit_balance, real_budget_spend, premium_account) values ('Mia1387', 'Mia', 'Yamamoto', 'Female', 'Student', 'Netherlands', 'Finance', '8:20:37', 125.32, 25.99, true);
cqlsh:chippi> SELECT * FROM userbase;

 user_id | account_type | credit_balance | first_name | gender | industry_of_choice | last_name | premium_account | real_budget_spend | region      | screen_time
---------+--------------+----------------+------------+--------+--------------------+-----------+-----------------+-------------------+-------------+-----------------
 Mia1387 |      Student |         125.32 |        Mia | Female |            Finance |  Yamamoto |            True |             25.99 | Netherlands | 08:20:37.000000000

(1 rows)
cqlsh:chippi>
```

```
(1 rows)
cqlsh:chippi> BEGIN BATCH
          ... INSERT INTO chippi(user_id, first_name, last_name, gender, account_type, region, industry_of_choice, screen_time, credit_balance, real_budget_spend, premium_account)
          ... values ('Logan3256', 'Logan', 'Wong', 'Male', 'Teacher', 'United Kingdom', 'Technology', '16:45:22', 15.99, 29.99, true);
          ... APPLY BATCH;
```

```
cqlsh:chippi>
cqlsh:chippi> INSERT INTO userbase (user_id, first_name, last_name, gender, account_type, region, industry_of_choice, screen_time, credit_balance, real_budget_spend, premium_account) values ('Noah2247', 'Noah', 'Kim', 'Male', 'Student', 'United Kingdom', 'Retail', '6:12:20', 65.99, 7.01, false);
cqlsh:chippi>
cqlsh:chippi> SELECT * FROM userbase;

 user_id   | account_type | credit_balance | first_name | gender | industry_of_choice | last_name | premium_account | real_budget_spend | region         | screen_time
-----------+--------------+----------------+------------+--------+--------------------+-----------+-----------------+-------------------+----------------+-----------------
   Mia1387 |      Student |         125.32 |        Mia | Female |            Finance |  Yamamoto |            True |             25.99 |    Netherlands | 08:20:37.000000000
  Noah2247 |      Student |          65.99 |       Noah |   Male |             Retail |       Kim |           False |              7.01 | United Kingdom | 06:12:20.000000000
 Logan3256 |      Teacher |          15.99 |      Logan |   Male |         Technology |      Wong |            True |             29.99 | United Kingdom | 16:45:22.000000000
```

```
cqlsh:chippi> SELECT * FROM userbase;

 user_id    | account_type | credit_balance | first_name | gender | industry_of_choice | last_name | premium_account | real_budget_spend | region         | screen_time
------------+--------------+----------------+------------+--------+--------------------+-----------+-----------------+-------------------+----------------+-----------------
    Mia1387 |      Student |         125.32 |        Mia | Female |            Finance |  Yamamoto |            True |             25.99 |    Netherlands | 08:20:37.000000000
   Noah2247 |      Student |          65.99 |       Noah |   Male |             Retail |       Kim |           False |              7.01 | United Kingdom | 06:12:20.000000000
 Daniel6009 |      Student |          77.54 |     Daniel |   Male |      Manufacturing |    Garcia |           False |            210.55 |          Spain | 04:24:15.000000000
  Logan3256 |      Teacher |          15.99 |      Logan |   Male |         Technology |      Wong |            True |             29.99 | United Kingdom | 16:45:22.000000000

(4 rows)
cqlsh:chippi> INSERT INTO userbase (user_id, first_name, last_name, gender, account_type, region, industry_of_choice, screen_time, credit_balance, real_budget_spend, premium_account) values ('Mersei2566', 'Mers
ei', 'Mongaba', 'Female', 'Student', 'United Kingdom', 'Aviation', '3:02:33', 67.22, 51.27, true);
cqlsh:chippi> INSERT INTO userbase (user_id, first_name, last_name, gender, account_type, region, industry_of_choice, screen_time, credit_balance, real_budget_spend, premium_account) values ('Adekola0156', 'Ade
kola', 'Adeniyi', 'Male', 'Student', 'Nigeria', 'Technology', '9:14:02', 23.21, 100.13, true);
cqlsh:chippi> INSERT INTO userbase (user_id, first_name, last_name, gender, account_type, region, industry_of_choice, screen_time, credit_balance, real_budget_spend, premium_account) values ('Caitlin1355', 'Cai
tlin', 'Nunez', 'Female', 'Teacher', 'United States', 'Law', '17:13:09', 33.91, 80.08, false);
cqlsh:chippi> INSERT INTO userbase (user_id, first_name, last_name, gender, account_type, region, industry_of_choice, screen_time, credit_balance, real_budget_spend, premium_account) values ('Maddie5677', 'Madd
ie', 'Madison', 'Female', 'Student', 'United Kingdom', 'Finance', '3:21:11', 4.95, 28.85, false);
cqlsh:chippi> INSERT INTO userbase (user_id, first_name, last_name, gender, account_type, region, industry_of_choice, screen_time, credit_balance, real_budget_spend, premium_account) values ('Nicole5677', 'Nico
le', 'Perez', 'Female', 'Student', 'Columbia', 'Technology', '8:17:16', 22.00, 56.89, true);
cqlsh:chippi> INSERT INTO userbase (user_id, first_name, last_name, gender, account_type, region, industry_of_choice, screen_time, credit_balance, real_budget_spend, premium_account) values ('Victoria0057', 'Vi
ctoria', 'Nelson', 'Female', 'Student', 'United Kingdom', 'Student', '5:14:05', 0.69, 87.21, false);
cqlsh:chippi> INSERT INTO userbase (user_id, first_name, last_name, gender, account_type, region, industry_of_choice, screen_time, credit_balance, real_budget_spend, premium_account) values ('Rachel0376', 'Rach
el', 'Adams', 'Female', 'Teacher', 'United States', 'Retail', '4:01:20', 2.56, 40.40, false);
cqlsh:chippi> SELECT * FROM userbase;
```

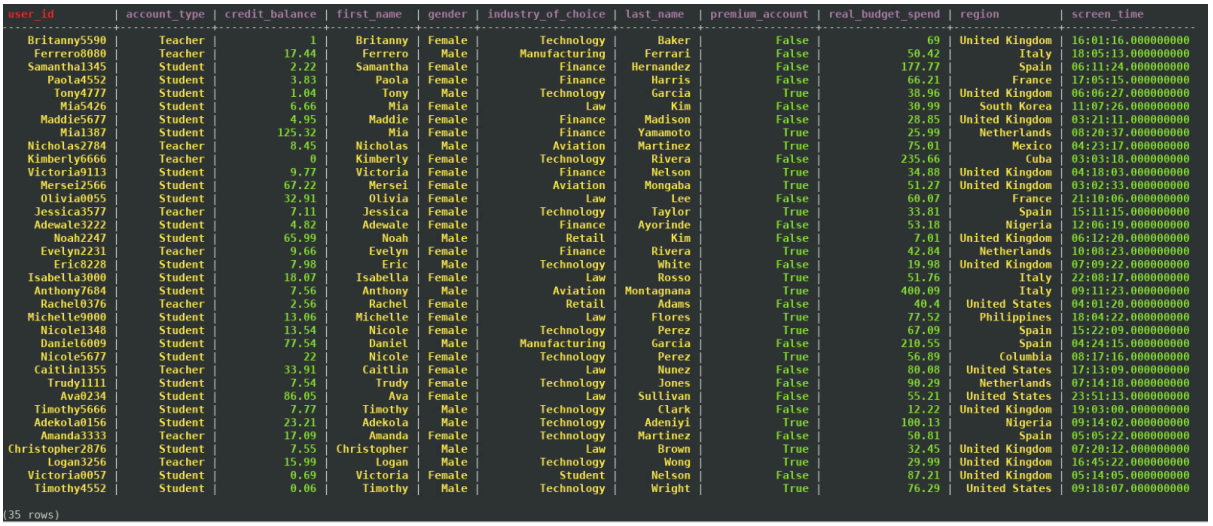
```
(11 rows)
cqlsh:chippi> INSERT INTO userbase (user_id, first_name, last_name, gender, account_type, region, industry_of_choice, screen_time, credit_balance, real_budget_spend, premium_account) values ('Tony4777', '
Tony', 'Garcia', 'Male', 'Student', 'United Kingdom', 'Technology', '6:06:27', 1.04, 38.96, true);
```

```
 user_id      | account_type | credit_balance | first_name | gender | industry_of_choice | last_name  | premium_account | real_budget_spend | region         | screen_time
--------------+--------------+----------------+------------+--------+--------------------+------------+-----------------+-------------------+----------------+-----------------
 Britanny5590 |      Teacher |              1 |   Britanny | Female |         Technology |      Baker |           False |                69 | United Kingdom | 16:01:16.000000000
 Samantha1345 |      Student |           2.22 |   Samantha | Female |            Finance |  Hernandez |           False |            177.77 |          Spain | 06:11:24.000000000
    Paola4552 |      Student |           3.83 |      Paola | Female |            Finance |     Harris |           False |             66.21 |         France | 17:05:15.000000000
     Tony4777 |      Student |           1.04 |       Tony |   Male |         Technology |     Garcia |            True |             38.96 | United Kingdom | 06:06:27.000000000
  Maddie5677 |      Student |           4.95 |     Maddie | Female |            Finance |    Madison |           False |             28.85 | United Kingdom | 03:21:11.000000000
    Mia1387 |      Student |         125.32 |        Mia | Female |            Finance |   Yamamoto |            True |             25.99 |    Netherlands | 08:20:37.000000000
 Kimberly6666 |      Teacher |              0 |   Kimberly | Female |         Technology |      Rivera |           False |            235.66 |           Cuba | 03:03:18.000000000
 Victoria9113 |      Student |           9.77 |   Victoria | Female |            Finance |     Nelson |            True |             34.88 | United Kingdom | 04:18:03.000000000
   Mersei2566 |      Student |          67.22 |     Mersei | Female |          Aviation |    Mongaba |            True |             51.27 | United Kingdom | 03:02:33.000000000
    Noah2247 |      Student |          65.99 |       Noah |   Male |             Retail |        Kim |           False |              7.01 | United Kingdom | 06:12:20.000000000
    Eric8228 |      Student |           7.98 |       Eric |   Male |         Technology |      White |           False |             19.98 | United Kingdom | 07:09:22.000000000
  Anthony7684 |      Student |           7.56 |    Anthony |   Male |          Aviation | Montagnana |            True |            400.09 |          Italy | 09:11:23.000000000
   Rachel0376 |      Teacher |           2.56 |     Rachel | Female |             Retail |      Adams |           False |              40.4 |  United States | 04:01:20.000000000
 Michelle9000 |      Student |          13.06 |   Michelle | Female |                Law |      Flores |            True |             77.52 |    Philippines | 18:04:22.000000000
   Daniel6009 |      Student |          77.54 |     Daniel |   Male |      Manufacturing |     Garcia |           False |            210.55 |          Spain | 04:24:15.000000000
    Nicole5677 |      Student |             22 |     Nicole | Female |         Technology |      Perez |            True |             56.89 |       Columbia | 08:17:16.000000000
  Caitlin1355 |      Teacher |          33.91 |    Caitlin | Female |                Law |      Nunez |           False |             80.08 |  United States | 17:13:09.000000000
  Adekola0156 |      Student |          23.21 |    Adekola |   Male |         Technology |    Adeniyi |            True |            100.13 |        Nigeria | 09:14:02.000000000
   Amanda3333 |      Teacher |          17.09 |     Amanda | Female |         Technology |   Martinez |           False |             50.81 |          Spain | 05:05:22.000000000
    Logan3256 |      Teacher |          15.99 |      Logan |   Male |         Technology |       Wong |            True |             29.99 | United Kingdom | 16:45:22.000000000
 Victoria0057 |      Student |           0.69 |   Victoria | Female |            Student |     Nelson |           False |             87.21 | United Kingdom | 05:14:05.000000000
  Timothy4552 |      Student |           0.06 |    Timothy |   Male |         Technology |     Wright |            True |             76.29 |  United States | 09:18:07.000000000

(22 rows)
```

```
cqlsh:chippi> INSERT INTO userbase (user_id, first_name, last_name, gender, account_type, region, industry_of_choice, screen_time, credit_balance, real_budget_spend, premium_account) values ('Amanda3333',
 'Amanda', 'Martinez', 'Female', 'Teacher', 'Spain', 'Technology', '5:05:22', 17.09, 50.81, false);
cqlsh:chippi> INSERT INTO userbase (user_id, first_name, last_name, gender, account_type, region, industry_of_choice, screen_time, credit_balance, real_budget_spend, premium_account) values ('Anthony7684'
, 'Anthony', 'Montagnana', 'Male', 'Student', 'Italy', 'Aviation', '9:11:23', 7.56, 400.09, true);
cqlsh:chippi> INSERT INTO userbase (user_id, first_name, last_name, gender, account_type, region, industry_of_choice, screen_time, credit_balance, real_budget_spend, premium_account) values ('Kimberly6666
', 'Kimberly', 'Rivera', 'Female', 'Teacher', 'Cuba', 'Technology', '3:03:18', 0.00, 235.66, false);
cqlsh:chippi> INSERT INTO userbase (user_id, first_name, last_name, gender, account_type, region, industry_of_choice, screen_time, credit_balance, real_budget_spend, premium_account) values ('Samantha1345
', 'Samantha', 'Hernandez', 'Female', 'Student', 'Spain', 'Finance', '6:11:24', 2.22, 177.77, false);
cqlsh:chippi> INSERT INTO userbase (user_id, first_name, last_name, gender, account_type, region, industry_of_choice, screen_time, credit_balance, real_budget_spend, premium_account) values ('Eric8228', '
Eric', 'White', 'Male', 'Student', 'United Kingdom', 'Technology', '7:09:22', 7.98, 19.98, false);
cqlsh:chippi> INSERT INTO userbase (user_id, first_name, last_name, gender, account_type, region, industry_of_choice, screen_time, credit_balance, real_budget_spend, premium_account) values ('Victoria9113
', 'Victoria', 'Nelson', 'Female', 'Student', 'United Kingdom', 'Finance', '4:18:03', 9.77, 34.88, true);
cqlsh:chippi> INSERT INTO userbase (user_id, first_name, last_name, gender, account_type, region, industry_of_choice, screen_time, credit_balance, real_budget_spend, premium_account) values ('Britanny5590
', 'Britanny', 'Baker', 'Female', 'Teacher', 'United Kingdom', 'Technology', '16:01:16', 1.00, 69.00, false);
cqlsh:chippi> INSERT INTO userbase (user_id, first_name, last_name, gender, account_type, region, industry_of_choice, screen_time, credit_balance, real_budget_spend, premium_account) values ('Michelle9000
', 'Michelle', 'Flores', 'Female', 'Student', 'Philippines', 'Law', '18:04:22', 13.06, 77.52, true);
cqlsh:chippi> INSERT INTO userbase (user_id, first_name, last_name, gender, account_type, region, industry_of_choice, screen_time, credit_balance, real_budget_spend, premium_account) values ('Timothy4552'
, 'Timothy', 'Wright', 'Male', 'Student', 'United States', 'Technology', '9:18:07', 0.06, 76.29, true);
cqlsh:chippi> INSERT INTO userbase (user_id, first_name, last_name, gender, account_type, region, industry_of_choice, screen_time, credit_balance, real_budget_spend, premium_account) values ('Paola4552',
'Paola', 'Harris', 'Female', 'Student', 'France', 'Finance', '17:05:15', 3.83, 66.21, false);
cqlsh:chippi> SELECT * FROM userbase;
```

```
cqlsh:chippi> INSERT INTO userbase (user_id, first_name, last_name, gender, account_type, region, industry_of_choice, screen_time, credit_balance, real_budget_spend, premium_account) values ('Trudy1111',
 'Trudy', 'Jones', 'Female', 'Student', 'Netherlands', 'Technology', '7:14:18', 7.54, 90.29, false);
cqlsh:chippi> INSERT INTO userbase (user_id, first_name, last_name, gender, account_type, region, industry_of_choice, screen_time, credit_balance, real_budget_spend, premium_account) values ('Nicholas278
4', 'Nicholas', 'Martinez', 'Male', 'Teacher', 'Mexico', 'Aviation', '4:23:17', 8.45, 75.01, true);
cqlsh:chippi> INSERT INTO userbase (user_id, first_name, last_name, gender, account_type, region, industry_of_choice, screen_time, credit_balance, real_budget_spend, premium_account) values ('Christopher
2876', 'Christopher', 'Brown', 'Male', 'Student', 'United Kingdom', 'Law', '7:20:12', 7.55, 32.45, true);
cqlsh:chippi> INSERT INTO userbase (user_id, first_name, last_name, gender, account_type, region, industry_of_choice, screen_time, credit_balance, real_budget_spend, premium_account) values ('Isabella300
0', 'Isabella', 'Rosso', 'Female', 'Student', 'Italy', 'Law', '22:08:17', 18.07, 51.76, true);
cqlsh:chippi> INSERT INTO userbase (user_id, first_name, last_name, gender, account_type, region, industry_of_choice, screen_time, credit_balance, real_budget_spend, premium_account) values ('Ferrero8080
', 'Ferrero', 'Ferrari', 'Male', 'Teacher', 'Italy', 'Manufacturing', '18:05:13', 17.44, 50.42, false);
cqlsh:chippi> SELECT * FROM userbase;
```

The command in the screenshot above showcases all the values within the userbase table which can be seen below:

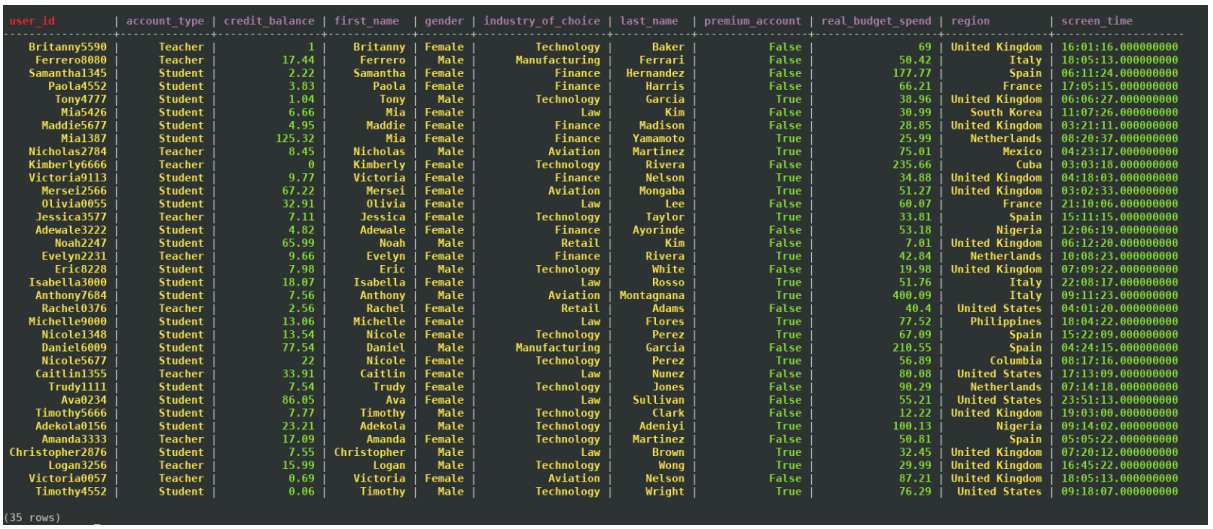


user_id	account_type	credit_balance	first_name	gender	industry_of_choice	last_name	premium_account	real_budget_spend	region	screen_time
Britanny5590	Teacher	1	Britanny	Female	Technology	Baker	False	69	United Kingdom	16:01:16.000000000
Ferrero8080	Teacher	17.44	Ferrero	Male	Manufacturing	Ferrari	False	50.42	Italy	18:05:13.000000000
Samantha1345	Student	2.22	Samantha	Female	Finance	Hernandez	False	177.77	Spain	06:11:24.000000000
Paola4552	Student	3.83	Paola	Female	Finance	Harris	False	66.21	France	17:05:15.000000000
Tony4777	Student	1.04	Tony	Male	Technology	Garcia	True	38.96	United Kingdom	06:06:27.000000000
Mia5426	Student	6.66	Mia	Female	Law	Kim	False	30.99	South Korea	11:07:26.000000000
Maddie5677	Student	4.95	Maddie	Female	Finance	Madison	False	28.85	United Kingdom	03:21:11.000000000
Mia1387	Student	125.32	Mia	Female	Finance	Yamamoto	True	25.99	Netherlands	08:20:37.000000000
Nicholas2784	Teacher	8.45	Nicholas	Male	Aviation	Martinez	True	75.01	Mexico	04:23:17.000000000
Kimberly6666	Teacher	0	Kimberly	Female	Technology	Rivera	False	235.66	Cuba	03:03:18.000000000
Victoria9113	Student	9.77	Victoria	Female	Finance	Nelson	True	34.88	United Kingdom	04:18:03.000000000
Mersei2566	Student	67.22	Mersei	Female	Aviation	Mongaba	True	51.27	United Kingdom	03:02:33.000000000
Olivia0055	Student	32.91	Olivia	Female	Law	Lee	False	60.07	France	21:10:06.000000000
Jessica3577	Teacher	7.11	Jessica	Female	Technology	Taylor	True	33.81	Spain	15:11:15.000000000
Adewale3222	Student	4.82	Adewale	Female	Finance	Ayorinde	False	53.18	Nigeria	12:06:19.000000000
Noah2247	Student	65.99	Noah	Male	Retail	Kim	False	7.01	United Kingdom	06:12:20.000000000
Evelyn2231	Teacher	9.66	Evelyn	Female	Finance	Rivera	True	42.84	Netherlands	10:08:23.000000000
Eric8228	Student	7.98	Eric	Male	Technology	White	False	19.98	United Kingdom	07:09:22.000000000
Isabella3000	Student	18.07	Isabella	Female	Law	Rosso	True	51.76	Italy	22:08:17.000000000
Anthony7684	Student	7.56	Anthony	Male	Aviation	Montagnana	True	400.09	Italy	09:11:23.000000000
Rachel0376	Teacher	2.56	Rachel	Female	Retail	Adams	False	40.4	United States	04:01:20.000000000
Michelle9000	Student	13.06	Michelle	Female	Law	Flores	True	77.52	Philippines	18:04:22.000000000
Nicole1348	Student	13.54	Nicole	Female	Technology	Perez	True	67.09	Spain	15:22:09.000000000
Daniel6009	Student	77.54	Daniel	Male	Manufacturing	Garcia	False	210.55	Spain	04:24:15.000000000
Nicole5677	Student	22	Nicole	Female	Technology	Perez	True	56.89	Columbia	08:17:16.000000000
Caitlin1355	Teacher	33.91	Caitlin	Female	Law	Nunez	False	80.08	United States	17:13:09.000000000
Trudy1111	Student	7.54	Trudy	Female	Technology	Jones	False	98.29	Netherlands	07:14:18.000000000
Ava0234	Student	86.05	Ava	Female	Law	Sullivan	False	55.21	United States	23:51:13.000000000
Timothy5666	Student	7.77	Timothy	Male	Technology	Clark	False	12.22	United Kingdom	19:03:00.000000000
Adekola0156	Student	23.21	Adekola	Male	Technology	Adeniyi	True	100.13	Nigeria	09:14:02.000000000
Amanda3333	Teacher	17.09	Amanda	Female	Technology	Martinez	False	50.81	Spain	05:05:22.000000000
Christopher2876	Student	7.55	Christopher	Male	Law	Brown	True	32.45	United Kingdom	07:20:12.000000000
Logan3256	Teacher	15.99	Logan	Male	Technology	Wong	True	29.99	United Kingdom	16:45:22.000000000
Victoria0057	Student	0.69	Victoria	Female	Student	Nelson	False	87.21	United Kingdom	05:14:05.000000000
Timothy4552	Student	0.06	Timothy	Male	Technology	Wright	True	76.29	United States	09:18:07.000000000

(35 rows)

This query updates the database to change user Victoria0057's industry of choice to Aviation:

```
cqlsh:chippi> UPDATE userbase
          ... SET industry_of_choice = 'Aviation'
          ... WHERE user_id = 'Victoria0057';
```



user_id	account_type	credit_balance	first_name	gender	industry_of_choice	last_name	premium_account	real_budget_spend	region	screen_time
Britanny5590	Teacher	1	Britanny	Female	Technology	Baker	False	69	United Kingdom	16:01:16.000000000
Ferrero8080	Teacher	17.44	Ferrero	Male	Manufacturing	Ferrari	False	50.42	Italy	18:05:13.000000000
Samantha1345	Student	2.22	Samantha	Female	Finance	Hernandez	False	177.77	Spain	06:11:24.000000000
Paola4552	Student	3.83	Paola	Female	Finance	Harris	False	66.21	France	17:05:15.000000000
Tony4777	Student	1.04	Tony	Male	Technology	Garcia	True	38.96	United Kingdom	06:06:27.000000000
Mia5426	Student	6.66	Mia	Female	Law	Kim	False	30.99	South Korea	11:07:26.000000000
Maddie5677	Student	4.95	Maddie	Female	Finance	Madison	False	28.85	United Kingdom	03:21:11.000000000
Mia1387	Student	125.32	Mia	Female	Finance	Yamamoto	True	25.99	Netherlands	08:20:37.000000000
Nicholas2784	Teacher	8.45	Nicholas	Male	Aviation	Martinez	True	75.01	Mexico	04:23:17.000000000
Kimberly6666	Teacher	0	Kimberly	Female	Technology	Rivera	False	235.66	Cuba	03:03:18.000000000
Victoria9113	Student	9.77	Victoria	Female	Finance	Nelson	True	34.88	United Kingdom	04:18:03.000000000
Mersei2566	Student	67.22	Mersei	Female	Aviation	Mongaba	True	51.27	United Kingdom	03:02:33.000000000
Olivia0055	Student	32.91	Olivia	Female	Law	Lee	False	60.07	France	21:10:06.000000000
Jessica3577	Teacher	7.11	Jessica	Female	Technology	Taylor	True	33.81	Spain	15:11:15.000000000
Adewale3222	Student	4.82	Adewale	Female	Finance	Ayorinde	False	53.18	Nigeria	12:06:19.000000000
Noah2247	Student	65.99	Noah	Male	Retail	Kim	False	7.01	United Kingdom	06:12:20.000000000
Evelyn2231	Teacher	9.66	Evelyn	Female	Finance	Rivera	True	42.84	Netherlands	10:08:23.000000000
Eric8228	Student	7.98	Eric	Male	Technology	White	False	19.98	United Kingdom	07:09:22.000000000
Isabella3000	Student	18.07	Isabella	Female	Law	Rosso	True	51.76	Italy	22:08:17.000000000
Anthony7684	Student	7.56	Anthony	Male	Aviation	Montagnana	True	400.09	Italy	09:11:23.000000000
Rachel0376	Teacher	2.56	Rachel	Female	Retail	Adams	False	40.4	United States	04:01:20.000000000
Michelle9000	Student	13.06	Michelle	Female	Law	Flores	True	77.52	Philippines	18:04:22.000000000
Nicole1348	Student	13.54	Nicole	Female	Technology	Perez	True	67.09	Spain	15:22:09.000000000
Daniel6009	Student	77.54	Daniel	Male	Manufacturing	Garcia	False	210.55	Spain	04:24:15.000000000
Nicole5677	Student	22	Nicole	Female	Technology	Perez	True	56.89	Columbia	08:17:16.000000000
Caitlin1355	Teacher	33.91	Caitlin	Female	Law	Nunez	False	80.08	United States	17:13:09.000000000
Trudy1111	Student	7.54	Trudy	Female	Technology	Jones	False	98.29	Netherlands	07:14:18.000000000
Ava0234	Student	86.05	Ava	Female	Law	Sullivan	False	55.21	United States	23:51:13.000000000
Timothy5666	Student	7.77	Timothy	Male	Technology	Clark	False	12.22	United Kingdom	19:03:00.000000000
Adekola0156	Student	23.21	Adekola	Male	Technology	Adeniyi	True	100.13	Nigeria	09:14:02.000000000
Amanda3333	Teacher	17.09	Amanda	Female	Technology	Martinez	False	50.81	Spain	05:05:22.000000000
Christopher2876	Student	7.55	Christopher	Male	Law	Brown	True	32.45	United Kingdom	07:20:12.000000000
Logan3256	Teacher	15.99	Logan	Male	Technology	Wong	True	29.99	United Kingdom	16:45:22.000000000
Victoria0057	Teacher	0.69	Victoria	Female	Aviation	Nelson	False	87.21	United Kingdom	18:05:13.000000000
Timothy4552	Student	0.06	Timothy	Male	Technology	Wright	True	76.29	United States	09:18:07.000000000

(35 rows)

An index was created for industry of choice to allow for more efficient data retrieval from queries:

```
cqlsh:chippi> CREATE INDEX on userbase(industry_of_choice);
```

This query calculates the sum of real budget spend for all users in the database and presents it with the title total_real_budget_spend.

```
cqlsh:chippi> SELECT SUM(real_budget_spend) AS total_real_budget_spend FROM userbase;

 total_real_budget_spend
-------------------------
                 2620.88

(1 rows)

Warnings :
Aggregation query used without partition key
```

This query selects all information of users that have technology as an industry of choice.

```
cqlsh:chippi> SELECT * FROM userbase WHERE industry_of_choice = 'Technology';

 user_id      | account_type | credit_balance | first_name | gender | industry_of_choice | last_name | premium_account | real_budget_spend | region         | screen_time
--------------+--------------+----------------+------------+--------+--------------------+-----------+-----------------+-------------------+----------------+-----------------
 Britanny5590 |      Teacher |              1 |   Britanny | Female |         Technology |     Baker |           False |                69 | United Kingdom | 16:01:16.000000000
     Tony4777 |      Student |           1.04 |       Tony |   Male |         Technology |    Garcia |            True |             38.96 | United Kingdom | 06:06:27.000000000
Kimberly6666 |      Teacher |              0 |   Kimberly | Female |         Technology |    Rivera |           False |            235.66 |           Cuba | 03:03:18.000000000
 Jessica3577 |      Teacher |           7.11 |    Jessica | Female |         Technology |    Taylor |            True |             33.81 |          Spain | 15:11:15.000000000
    Eric8228 |      Student |           7.98 |       Eric |   Male |         Technology |     White |           False |             19.98 | United Kingdom | 07:09:22.000000000
  Nicole1348 |      Student |          13.54 |      Nicole | Female |         Technology |     Perez |            True |             67.09 |          Spain | 15:22:09.000000000
  Nicole5677 |      Student |             22 |      Nicole | Female |         Technology |     Perez |            True |             56.89 |       Columbia | 08:17:16.000000000
   Trudy1111 |      Student |           7.54 |      Trudy | Female |         Technology |     Jones |           False |             90.29 |    Netherlands | 07:14:18.000000000
 Timothy5666 |      Student |           7.77 |    Timothy |   Male |         Technology |     Clark |           False |             12.22 | United Kingdom | 19:03:00.000000000
 Adekola0156 |      Student |          23.21 |    Adekola |   Male |         Technology |   Adeniyi |            True |            100.13 |        Nigeria | 09:14:02.000000000
  Amanda3333 |      Teacher |          17.09 |     Amanda | Female |         Technology |  Martinez |           False |             50.81 |          Spain | 05:05:22.000000000
   Logan3256 |      Teacher |          15.99 |      Logan |   Male |         Technology |      Wong |            True |             29.99 | United Kingdom | 16:45:22.000000000
 Timothy4552 |      Student |           0.06 |    Timothy |   Male |         Technology |    Wright |            True |             76.29 |  United States | 09:18:07.000000000

(13 rows)
```

This copies the data of the database ‘userbase’ into a newly created table called ‘updated_userbase’:

```
cqlsh:chippi> COPY chippi.userbase (user_id, account_type, credit_balance, first_name, gender, industry_of_choice, last_name, premium_account, real_budget_spend, region) TO 'temp.csv';
Using 1 child processes

Starting copy of chippi.userbase with columns [user_id, account_type, credit_balance, first_name, gender, industry_of_choice, last_name, premium_account, real_budget_spend, region].
Processed: 35 rows; Rate:      69 rows/s; Avg. rate:      68 rows/s
35 rows exported to 1 files in 0.522 seconds.
cqlsh:chippi> COPY chippi.updated_userbase (user_id, account_type, credit_balance, first_name, gender, industry_of_choice, last_name, premium_account, real_budget_spend, region) FROM 'temp.csv';
Using 1 child processes

Starting copy of chippi.updated_userbase with columns [user_id, account_type, credit_balance, first_name, gender, industry_of_choice, last_name, premium_account, real_budget_spend, region].
Processed: 35 rows; Rate:      65 rows/s; Avg. rate:      95 rows/s
35 rows imported from 1 files in 0.369 seconds (0 skipped).
cqlsh:chippi>
cqlsh:chippi> SELECT * FROM updated_userbase;
```

```
 user_id      | account_type | credit_balance | first_name | gender | industry_of_choice | last_name | premium_account | real_budget_spend | region         | screen_time
--------------+--------------+----------------+------------+--------+--------------------+-----------+-----------------+-------------------+----------------+-------------
 Britanny5590 |      Teacher |              1 |   Britanny | Female |         Technology |     Baker |           False |                69 | United Kingdom |        null
    Ferrero8080 |      Teacher |          17.44 |    Ferrero |   Male |      Manufacturing |    Ferrari |           False |             50.42 |          Italy |        null
  Samantha1345 |      Student |           2.22 |   Samantha | Female |            Finance |  Hernandez |           False |            177.77 |          Spain |        null
     Paola4552 |      Student |           3.83 |      Paola | Female |            Finance |    Harris |           False |             66.21 |         France |        null
     Tony4777 |      Student |           1.04 |       Tony |   Male |         Technology |    Garcia |            True |             38.96 | United Kingdom |        null
      Mia5426 |      Student |           6.66 |        Mia | Female |                Law |       Kim |           False |             30.99 |    South Korea |        null
  Maddie5677 |      Student |           4.95 |     Maddie | Female |            Finance |    Madison |           False |             28.85 | United Kingdom |        null
     Mia1387 |      Student |         125.32 |        Mia | Female |            Finance |   Yamamoto |            True |             25.99 |    Netherlands |        null
  Nicholas2784 |      Teacher |           8.45 |    Nicholas |   Male |          Aviation |  Martinez |            True |             75.01 |         Mexico |        null
  Kimberly6666 |      Teacher |              0 |   Kimberly | Female |         Technology |    Rivera |           False |            235.66 |           Cuba |        null
  Victoria9113 |      Student |           9.77 |   Victoria | Female |            Finance |    Nelson |            True |             34.88 | United Kingdom |        null
    Mersei2566 |      Student |          67.22 |     Mersei | Female |          Aviation |   Mongaba |            True |             51.27 | United Kingdom |        null
   Olivia0055 |      Student |          32.91 |     Olivia | Female |                Law |       Lee |           False |             60.07 |         France |        null
   Jessica3577 |      Teacher |           7.11 |    Jessica | Female |         Technology |    Taylor |            True |             33.81 |          Spain |        null
   Adewale3222 |      Student |           4.82 |    Adewale | Female |            Finance |  Ayorinde |           False |             53.18 |        Nigeria |        null
     Noah2247 |      Student |          65.99 |       Noah |   Male |             Retail |       Kim |           False |              7.01 | United Kingdom |        null
   Evelyn2231 |      Teacher |           9.66 |     Evelyn | Female |            Finance |    Rivera |            True |             42.84 |    Netherlands |        null
    Eric8228 |      Student |           7.98 |       Eric |   Male |         Technology |     White |           False |             19.98 | United Kingdom |        null
  Isabella3000 |      Student |          18.07 |   Isabella | Female |                Law |     Rosso |            True |             51.76 |          Italy |        null
   Anthony7684 |      Student |           7.56 |    Anthony |   Male |          Aviation | Montagnana |            True |            400.09 |          Italy |        null
   Rachel0376 |      Teacher |           2.56 |     Rachel | Female |             Retail |     Adams |           False |              40.4 |  United States |        null
  Michelle9000 |      Student |          13.06 |    Michelle | Female |                Law |     Flores |            True |             77.52 |    Philippines |        null
    Nicole1348 |      Student |          13.54 |      Nicole | Female |         Technology |     Perez |            True |             67.09 |          Spain |        null
    Daniel6009 |      Student |          77.54 |     Daniel |   Male |      Manufacturing |    Garcia |           False |            210.55 |          Spain |        null
    Nicole5677 |      Student |             22 |      Nicole | Female |         Technology |     Perez |            True |             56.89 |       Columbia |        null
   Caitlin1355 |      Teacher |          33.91 |    Caitlin | Female |                Law |     Nunez |           False |             80.08 |  United States |        null
     Trudy1111 |      Student |           7.54 |      Trudy | Female |         Technology |     Jones |           False |             90.29 |    Netherlands |        null
      Ava0234 |      Student |          86.05 |        Ava | Female |                Law |  Sullivan |           False |             55.21 |  United States |        null
   Timothy5666 |      Student |           7.77 |    Timothy |   Male |         Technology |     Clark |           False |             12.22 | United Kingdom |        null
   Adekola0156 |      Student |          23.21 |    Adekola |   Male |         Technology |   Adeniyi |            True |            100.13 |        Nigeria |        null
    Amanda3333 |      Teacher |          17.09 |     Amanda | Female |         Technology |  Martinez |           False |             50.81 |          Spain |        null
Christopher2876 |      Teacher |           7.55 | Christopher |   Male |                Law |     Brown |            True |             32.45 | United Kingdom |        null
     Logan3256 |      Teacher |          15.99 |      Logan |   Male |         Technology |      Wong |            True |             29.99 | United Kingdom |        null
   Victoria0057 |      Teacher |           0.69 |   Victoria | Female |          Aviation |    Nelson |           False |             87.21 | United Kingdom |        null
   Timothy4552 |      Student |           0.06 |    Timothy |   Male |         Technology |    Wright |            True |             76.29 |  United States |        null

(35 rows)
cqlsh:chippi>
```

Creation of new table main_userbase, note that screen time has been listed as a floating number as this is a more appropriate format for the data. Though varchar and text formats can be used interchangeably, text is used in this table as I cannot foresee a need for text limitation.

```
cqlsh:chippi> CREATE TABLE chippi.main_userbase (
          ...             user_id text PRIMARY KEY,
          ...             account_type text,
          ...             credit_balance double,
          ...             first_name text,
          ...             gender text,
          ...             industry_of_choice text,
          ...             last_name text,
          ...             premium_account boolean,
          ...             real_budget_spend double,
          ...             region text,
          ...             screen_time float
          ... );
cqlsh:chippi>
```

```
cqlsh:chippi> ALTER TABLE chippi.main_userbase
          ... ADD password text;
```

Selected necessary labels created for a relatively optimal table:

```
cqlsh:chippi> SELECT * FROM main_userbase;

 user_id | account_type | credit_balance | first_name | gender | industry_of_choice | last_name | password | premium_account | real_budget_spend | region | screen_time
---------+--------------+----------------+------------+--------+--------------------+-----------+----------+-----------------+-------------------+--------+-------------
```

Data copied from old to new table:

```
cqlsh:chippi> COPY chippi.updated_userbase (user_id, account_type, credit_balance, first_name, gender, industry_of_choice, last_name, premium_account, real_budget_spend, region) TO 'temp.csv';
Using 1 child processes

Starting copy of chippi.updated_userbase with columns [user_id, account_type, credit_balance, first_name, gender, industry_of_choice, last_name, premium_account, real_budget_spend, region].
Processed: 35 rows; Rate:      87 rows/s; Avg. rate:      79 rows/s
35 rows exported to 1 files in 0.446 seconds.
cqlsh:chippi> COPY chippi.main_userbase (user_id, account_type, credit_balance, first_name, gender, industry_of_choice, last_name, premium_account, real_budget_spend, region) FROM 'temp.csv';
Using 1 child processes

Starting copy of chippi.main_userbase with columns [user_id, account_type, credit_balance, first_name, gender, industry_of_choice, last_name, premium_account, real_budget_spend, region].
Processed: 35 rows; Rate:      63 rows/s; Avg. rate:      92 rows/s
35 rows imported from 1 files in 0.379 seconds (0 skipped).
cqlsh:chippi>
```

Newly created main_userbase table including data:

```
 user_id          | account_type | credit_balance | first_name  | gender | industry_of_choice | last_name  | password | premium_account | real_budget_spend | region         | screen_time
------------------+--------------+----------------+-------------+--------+--------------------+------------+----------+-----------------+-------------------+----------------+-------------
    Britanny5590  |      Teacher |              1 |    Britanny | Female |         Technology |      Baker |     null |           False |                69 | United Kingdom |        null
     Ferrero8080  |      Teacher |          17.44 |     Ferrero |   Male |      Manufacturing |    Ferrari |     null |           False |             50.42 |          Italy |        null
   Samantha1345   |      Student |           2.22 |    Samantha | Female |            Finance |  Hernandez |     null |           False |            177.77 |          Spain |        null
      Paola4552   |      Student |           3.83 |       Paola | Female |            Finance |     Harris |     null |           False |             66.21 |         France |        null
       Tony4777   |      Student |           1.04 |        Tony |   Male |         Technology |     Garcia |     null |            True |             38.96 | United Kingdom |        null
       Mia5426    |      Student |           6.66 |         Mia | Female |                Law |        Kim |     null |           False |             30.99 |    South Korea |        null
     Maddie5677   |      Student |           4.95 |      Maddie | Female |            Finance |    Madison |     null |           False |             28.85 | United Kingdom |        null
      Mia1387     |      Student |         125.32 |         Mia | Female |            Finance |   Yamamoto |     null |            True |             25.99 |    Netherlands |        null
   Nicholas2784   |      Teacher |           8.45 |    Nicholas |   Male |          Aviation |   Martinez |     null |            True |             75.01 |         Mexico |        null
   Kimberly6666   |      Teacher |              0 |     Kimberly | Female |         Technology |      Rivera |     null |           False |            235.66 |           Cuba |        null
   Victoria9113   |      Student |           9.77 |     Victoria | Female |            Finance |     Nelson |     null |            True |             34.88 | United Kingdom |        null
     Mersei2566   |      Student |          67.22 |       Mersei | Female |          Aviation |    Mongaba |     null |            True |             51.27 | United Kingdom |        null
     Olivia0055   |      Student |          32.91 |       Olivia | Female |                Law |        Lee |     null |           False |             60.07 |         France |        null
    Jessica3577   |      Teacher |           7.11 |      Jessica | Female |         Technology |     Taylor |     null |            True |             33.81 |          Spain |        null
    Adewale3222   |      Student |           4.82 |      Adewale | Female |            Finance |   Ayorinde |     null |           False |             53.18 |        Nigeria |        null
      Noah2247    |      Student |          65.99 |        Noah |   Male |             Retail |        Kim |     null |           False |              7.01 | United Kingdom |        null
    Evelyn2231    |      Teacher |           9.66 |      Evelyn | Female |            Finance |      Rivera |     null |            True |             42.84 |    Netherlands |        null
     Eric8228     |      Student |           7.98 |        Eric |   Male |         Technology |      White |     null |           False |             19.98 | United Kingdom |        null
   Isabella3000   |      Student |          18.07 |    Isabella | Female |                Law |      Rosso |     null |            True |             51.76 |          Italy |        null
    Anthony7684   |      Student |           7.56 |     Anthony |   Male |          Aviation | Montagnana |     null |            True |            400.09 |          Italy |        null
     Rachel8376   |      Teacher |           2.56 |      Rachel | Female |             Retail |      Adams |     null |           False |             40.4  |  United States |        null
   Michelle9000   |      Student |          13.06 |    Michelle | Female |                Law |      Flores |     null |            True |             77.52 |    Philippines |        null
     Nicole1348   |      Student |          13.54 |       Nicole | Female |         Technology |      Perez |     null |            True |             67.09 |          Spain |        null
     Daniel6009   |      Student |          77.54 |      Daniel |   Male |      Manufacturing |     Garcia |     null |           False |            210.55 |          Spain |        null
     Nicole5677   |      Student |             22 |       Nicole | Female |         Technology |      Perez |     null |            True |             56.89 |       Columbia |        null
    Caitlin1355   |      Teacher |          33.91 |     Caitlin | Female |                Law |      Nunez |     null |           False |             80.08 |  United States |        null
      Trudy1111   |      Student |           7.54 |       Trudy | Female |         Technology |      Jones |     null |           False |             90.29 |    Netherlands |        null
       Ava0234    |      Student |          86.05 |         Ava | Female |                Law |   Sullivan |     null |           False |             55.21 |  United States |        null
    Timothy5666   |      Student |           7.77 |     Timothy |   Male |         Technology |      Clark |     null |           False |             12.22 | United Kingdom |        null
    Adekola0156   |      Student |          23.21 |     Adekola |   Male |         Technology |     Adeniyi |     null |            True |            100.13 |        Nigeria |        null
    Amanda3333    |      Teacher |          17.09 |      Amanda | Female |         Technology |   Martinez |     null |           False |             50.81 |          Spain |        null
 Christopher2876  |      Student |           7.55 | Christopher |   Male |                Law |      Brown |     null |            True |             32.45 | United Kingdom |        null
     Logan3256    |      Teacher |          15.99 |       Logan |   Male |         Technology |       Wong |     null |            True |             29.99 | United Kingdom |        null
    Victoria0057  |      Teacher |           0.69 |     Victoria | Female |          Aviation |     Nelson |     null |           False |             87.21 | United Kingdom |        null
    Timothy4552   |      Student |           0.06 |     Timothy |   Male |         Technology |     Wright |     null |            True |             76.29 |  United States |        null

(35 rows)
cqlsh:chippi>
```

Batch data input (password and screen time):

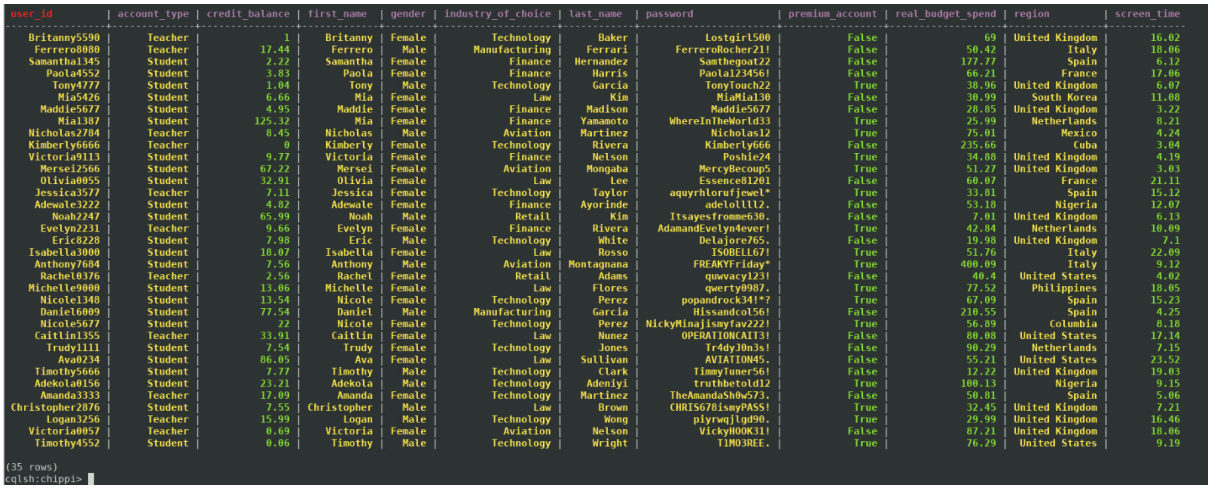
```
cqlsh:chippi> UPDATE main_userbase SET password = 'Lostgirl500' WHERE user_id = 'Britanny5590';
cqlsh:chippi> UPDATE main_userbase SET password = 'FerreroRocher21!' WHERE user_id = 'Ferrero8080';
cqlsh:chippi> UPDATE main_userbase SET password = 'Samthegoat22' WHERE user_id = 'Samantha1345';
cqlsh:chippi> UPDATE main_userbase SET password = 'Paola123456!' WHERE user_id = 'Paola4552';
```

```
cqlsh:chippi> BEGIN BATCH
         ... UPDATE main_userbase SET password = 'TonyTouch22' WHERE user_id = 'Tony4777';
         ... UPDATE main_userbase SET password = 'MiaMia130' WHERE user_id = 'Mia5426';
         ... UPDATE main_userbase SET password = 'Maddie5677' WHERE user_id = 'Maddie5677';
         ... UPDATE main_userbase SET password = 'WhereInTheWorld33' WHERE user_id = 'Mia1387';
         ... UPDATE main_userbase SET password = 'Nicholas12' WHERE user_id = 'Nicholas2784';
         ... UPDATE main_userbase SET password = 'Kimberly666' WHERE user_id = 'Kimberly6666';
         ... UPDATE main_userbase SET password = 'Poshie24' WHERE user_id = 'Victoria9113';
         ... UPDATE main_userbase SET password = 'MercyBecoup5' WHERE user_id = 'Mersei2566';
         ... UPDATE main_userbase SET password = 'Essence81201' WHERE user_id = 'Olivia0055';
         ... APPLY BATCH;
cqlsh:chippi> BEGIN BATCH
         ... UPDATE main_userbase SET password = 'aquyrhlorufjewel*' WHERE user_id = 'Jessica3577';
         ... UPDATE main_userbase SET password = 'adelolllll2.' WHERE user_id = 'Adewale3222';
         ... UPDATE main_userbase SET password = 'Itsayesfromme630.' WHERE user_id = 'Noah2247';
         ... UPDATE main_userbase SET password = 'AdamandEvelyn4ever!' WHERE user_id = 'Evelyn2231';
         ... UPDATE main_userbase SET password = 'Delajore765.' WHERE user_id = 'Eric8228';
         ... UPDATE main_userbase SET password = 'ISOBELL67!' WHERE user_id = 'Isabella3000';
         ... UPDATE main_userbase SET password = 'FREAKYFriday*' WHERE user_id = 'Anthony7684';
         ... UPDATE main_userbase SET password = 'quwvacy123!' WHERE user_id = 'Rachel0376';
         ... UPDATE main_userbase SET password = 'qwerty0987.' WHERE user_id = 'Michelle9000';
         ... UPDATE main_userbase SET password = 'popandrock34!*?' WHERE user_id = 'Nicole1348';
         ... UPDATE main_userbase SET password = 'Hissandcol56!' WHERE user_id = 'Daniel6009';
         ... UPDATE main_userbase SET password = 'NickyMinajismyfav222!' WHERE user_id = 'Nicole5677';
         ... UPDATE main_userbase SET password = 'OPERATIONCAIT3!' WHERE user_id = 'Caitlin1355';
         ... UPDATE main_userbase SET password = 'Tr4dyJ0n3s!' WHERE user_id = 'Trudy1111';
         ... UPDATE main_userbase SET password = 'AVIATION45.' WHERE user_id = 'Ava0234';
         ... UPDATE main_userbase SET password = 'TimmyTuner56!' WHERE user_id = 'Timothy5666';
         ... UPDATE main_userbase SET password = 'truthbetold12' WHERE user_id = 'Adekola0156';
         ... UPDATE main_userbase SET password = 'TheAmandaSh0w573.' WHERE user_id = 'Amanda3333';
         ... UPDATE main_userbase SET password = 'CHRIS678ismyPASS!' WHERE user_id = 'Christopher2876';
         ... UPDATE main_userbase SET password = 'piyrwqjlgd90.' WHERE user_id = 'Logan3256';
         ... UPDATE main_userbase SET password = 'VickyHOOK31!' WHERE user_id = 'Victoria0057';
         ... UPDATE main_userbase SET password = 'T1MO3REE.' WHERE user_id = 'Timothy4552';
         ... APPLY BATCH;
cqlsh:chippi>
```

```
cqlsh:chippi> UPDATE main_userbase SET screen_time = 16.02 WHERE user_id = 'Britanny5590';
cqlsh:chippi> UPDATE main_userbase SET screen_time = 18.06 WHERE user_id = 'Ferrero8080';
cqlsh:chippi> UPDATE main_userbase SET screen_time = 6.12 WHERE user_id = 'Samantha1345';
cqlsh:chippi> UPDATE main_userbase SET screen_time = 17.06 WHERE user_id = 'Paola4552';
```

```
cqlsh:chippi> UPDATE main_userbase SET screen_time = 19.03 WHERE user_id = 'Timothy5666';
cqlsh:chippi> UPDATE main_userbase SET screen_time = 9.15 WHERE user_id = 'Adekola0156';
cqlsh:chippi> UPDATE main_userbase SET screen_time = 5.06 WHERE user_id = 'Amanda3333';
cqlsh:chippi> UPDATE main_userbase SET screen_time = 7.21 WHERE user_id = 'Christopher2876';
cqlsh:chippi> UPDATE main_userbase SET screen_time = 16.46 WHERE user_id = 'Logan3256';
cqlsh:chippi> UPDATE main_userbase SET screen_time = 18.06 WHERE user_id = 'Victoria0057';
cqlsh:chippi> UPDATE main_userbase SET screen_time = 9.19 WHERE user_id = 'Timothy4552';
cqlsh:chippi>
```

```
cqlsh:chippi> UPDATE main_userbase SET screen_time = 6.07 WHERE user_id = 'Tony4777';
cqlsh:chippi> UPDATE main_userbase SET screen_time = 11.08 WHERE user_id = 'Mia5426';
cqlsh:chippi> UPDATE main_userbase SET screen_time = 3.22 WHERE user_id = 'Maddie5677';
cqlsh:chippi> UPDATE main_userbase SET screen_time = 8.21 WHERE user_id = 'Mia1387';
cqlsh:chippi> UPDATE main_userbase SET screen_time = 4.24 WHERE user_id = 'Nicholas2784';
cqlsh:chippi> UPDATE main_userbase SET screen_time = 3.04 WHERE user_id = 'Kimberly6666';
cqlsh:chippi> UPDATE main_userbase SET screen_time = 4.19 WHERE user_id = 'Victoria9113';
cqlsh:chippi> UPDATE main_userbase SET screen_time = 3.03 WHERE user_id = 'Mersei2566';
cqlsh:chippi> UPDATE main_userbase SET screen_time = 21.11 WHERE user_id = 'Olivia0055';
cqlsh:chippi> UPDATE main_userbase SET screen_time = 15.12 WHERE user_id = 'Jessica3577';
cqlsh:chippi> UPDATE main_userbase SET screen_time = 12.07 WHERE user_id = 'Adewale3222';
cqlsh:chippi> UPDATE main_userbase SET screen_time = 6.13 WHERE user_id = 'Noah2247';
cqlsh:chippi> UPDATE main_userbase SET screen_time = 10.09 WHERE user_id = 'Evelyn2231';
cqlsh:chippi> UPDATE main_userbase SET screen_time = 7.10 WHERE user_id = 'Eric8228';
cqlsh:chippi> UPDATE main_userbase SET screen_time = 22.09 WHERE user_id = 'Isabella3000';
cqlsh:chippi> UPDATE main_userbase SET screen_time = 9.12 WHERE user_id = 'Anthony7684';
cqlsh:chippi> UPDATE main_userbase SET screen_time = 4.02 WHERE user_id = 'Rachel0376';
cqlsh:chippi> UPDATE main_userbase SET screen_time = 18.05 WHERE user_id = 'Michelle9000';
cqlsh:chippi> UPDATE main_userbase SET screen_time = 15.23 WHERE user_id = 'Nicole1348';
cqlsh:chippi> UPDATE main_userbase SET screen_time = 4.25 WHERE user_id = 'Daniel6009';
cqlsh:chippi> UPDATE main_userbase SET screen_time = 8.18 WHERE user_id = 'Nicole5677';
cqlsh:chippi> UPDATE main_userbase SET screen_time = 17.14 WHERE user_id = 'Caitlin1355';
cqlsh:chippi> UPDATE main_userbase SET screen_time = 7.15 WHERE user_id = 'Trudy1111';
cqlsh:chippi> UPDATE main_userbase SET screen_time = 23.52 WHERE user_id = 'Ava0234';
cqlsh:chippi>
```



user_id	account_type	credit_balance	first_name	gender	industry_of_choice	last_name	password	premium_account	real_budget_spend	region	screen_time
Britanny5590	Teacher	1	Britanny	Female	Technology	Baker	Lostgirl500	False	69	United Kingdom	16.02
Ferrero8080	Teacher	17.44	Ferrero	Male	Manufacturing	Ferrari	FerreroRocher21!	False	50.42	Italy	18.06
Samanthal345	Student	2.22	Samantha	Female	Finance	Hernandez	Samthegoat22	False	177.77	Spain	6.12
Paola4552	Student	3.83	Paola	Female	Finance	Harris	Paola123456!	False	66.21	France	17.06
Tony4777	Student	1.04	Tony	Male	Technology	Garcia	TonyTouch22	True	38.96	United Kingdom	6.07
Mia5426	Student	6.66	Mia	Female	Law	Kim	MiaMia130	False	30.99	South Korea	11.08
Maddie5677	Student	4.95	Maddie	Female	Finance	Madison	Maddie5677	False	20.85	United Kingdom	3.22
Mia1307	Student	125.32	Mia	Female	Finance	Yamamoto	WhereInTheWorld33	True	25.99	Netherlands	8.21
Nicholas2784	Teacher	8.45	Nicholas	Male	Aviation	Martinez	Nicholas12	True	75.01	Mexico	4.24
Kimberly6666	Teacher	0	Kimberly	Female	Technology	Rivera	Kimberly666	False	235.66	Cuba	3.04
Victoria9113	Student	9.77	Victoria	Female	Finance	Nelson	Poshie24	True	34.88	United Kingdom	4.19
Mersei2566	Student	67.22	Mersei	Female	Aviation	Mongaba	MercyBecoup5	True	51.27	United Kingdom	3.03
Olivia0055	Student	32.91	Olivia	Female	Law	Lee	Essence81201	False	60.07	France	21.11
Jessica3577	Teacher	7.11	Jessica	Female	Technology	Taylor	aquyrhlorufjewel*	True	33.81	Spain	15.12
Adewale3222	Student	4.82	Adewale	Female	Finance	Ayorinde	adelolll12.	False	53.18	Nigeria	12.07
Noah2247	Student	65.99	Noah	Male	Retail	Kim	Itsayesfromme630.	False	7.01	United Kingdom	6.13
Evelyn2231	Teacher	9.66	Evelyn	Female	Finance	Rivera	AdamandEvelyn4ever!	True	42.84	Netherlands	10.09
Eric8228	Student	7.98	Eric	Male	Technology	White	Delajore765.	False	19.98	United Kingdom	7.1
Isabella3000	Student	10.07	Isabella	Female	Law	Rosso	ISOBELL67!	True	51.76	Italy	22.09
Anthony7684	Student	7.56	Anthony	Male	Aviation	Montagnana	FREAKYfriday*	True	400.09	Italy	9.12
Rachel0376	Teacher	2.56	Rachel	Female	Retail	Adams	quwvacyl23!	False	40.4	United States	4.02
Michelle9000	Student	13.06	Michelle	Female	Law	Flores	qwerty9087.	True	77.52	Philippines	18.05
Nicole1348	Student	13.54	Nicole	Female	Technology	Perez	popandrock34!*?	True	67.09	Spain	15.23
Daniel6009	Student	77.54	Daniel	Male	Manufacturing	Garcia	Hissandcol56!	False	210.55	Spain	4.25
Nicole5677	Student	22	Nicole	Female	Technology	Perez	NickyMinajismyfav222!	True	56.89	Columbia	8.18
Caitlin1355	Teacher	33.91	Caitlin	Female	Law	Nunez	OPERATIONCAIT3!	False	80.08	United States	17.14
Trudy1111	Student	7.54	Trudy	Female	Technology	Jones	Tr4dyJ0n3s!	False	90.29	Netherlands	7.15
Ava0234	Student	86.05	Ava	Female	Law	Sullivan	AVIATION45.	False	55.21	United States	23.52
Timothy5666	Student	7.77	Timothy	Male	Technology	Clark	TimmyTuner56!	False	12.22	United Kingdom	19.03
Adekola0156	Student	23.21	Adekola	Male	Technology	Adeniyi	truthbetold12	True	100.13	Nigeria	9.15
Amanda3333	Teacher	17.09	Amanda	Female	Technology	Martinez	TheAmandaSh0w573.	False	50.81	Spain	5.06
Christopher2876	Student	7.55	Christopher	Male	Law	Brown	CHRIS678ismyPASS!	True	32.45	United Kingdom	7.21
Logan3256	Teacher	15.99	Logan	Male	Technology	Wong	plyrwqjlqd90.	True	29.99	United Kingdom	16.46
Victoria0057	Teacher	0.69	Victoria	Female	Aviation	Nelson	VickyHOOK31!	False	87.21	United Kingdom	18.06
Timothy4552	Student	0.06	Timothy	Male	Technology	Wright	TIMO3REE.	True	76.29	United States	9.19

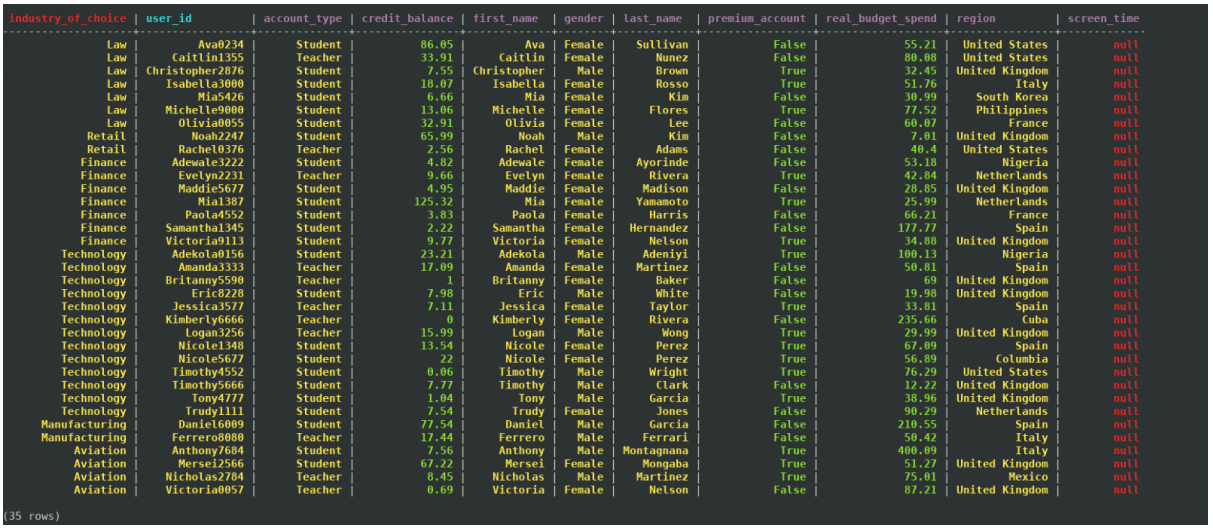
(35 rows)
cqlsh:chippi>

I created a duplicate table and added two columns under primary key to run a query that 'groups' the column:

```
cqlsh:chippi> CREATE TABLE chippi.duplicate_userbase (
          ...           user_id text,
          ...           account_type text,
          ...           credit_balance double,
          ...           first_name text,
          ...           gender text,
          ...           industry_of_choice text,
          ...           last_name text,
          ...           premium_account boolean,
          ...           real_budget_spend double,
          ...           region text,
          ...           screen_time float,
          ...           PRIMARY KEY (industry_of_choice, user_id)
          ... );
cqlsh:chippi> SELECT
          ... industry_of_choice,
          ... AVG(real_budget_spend) AS average_spend
          ... FROM
          ... chippi.duplicate_userbase
          ... GROUP BY industry_of_choice;

 industry_of_choice | average_spend
--------------------+---------------

(0 rows)

Warnings :
Aggregation query used without partition key

cqlsh:chippi>
```

Duplicate table with same data as main:

```
cqlsh:chippi> COPY chippi.main_userbase (user_id, account_type, credit_balance, first_name, gender, industry_of_choice, last_name, premium_account, real_budget_spend, region) TO 'temp.csv';
Using 1 child processes

Starting copy of chippi.main_userbase with columns [user_id, account_type, credit_balance, first_name, gender, industry_of_choice, last_name, premium_account, real_budget_spend, region].
Processed: 35 rows; Rate:      100 rows/s; Avg. rate:      88 rows/s
35 rows exported to 1 files in 0.403 seconds.
cqlsh:chippi> COPY chippi.duplicate_userbase (user_id, account_type, credit_balance, first_name, gender, industry_of_choice, last_name, premium_account, real_budget_spend, region) FROM 'temp.csv';
Using 1 child processes

Starting copy of chippi.duplicate_userbase with columns [user_id, account_type, credit_balance, first_name, gender, industry_of_choice, last_name, premium_account, real_budget_spend, region].
Processed: 35 rows; Rate:      58 rows/s; Avg. rate:      87 rows/s
35 rows imported from 1 files in 0.404 seconds (0 skipped).
cqlsh:chippi> SELECT * FROM duplicate_userbase;
```



industry_of_choice	user_id	account_type	credit_balance	first_name	gender	last_name	premium_account	real_budget_spend	region	screen_time
Law	Ava0234	Student	86.05	Ava	Female	Sullivan	False	55.21	United States	null
Law	Caitlin1355	Teacher	33.91	Caitlin	Female	Nunez	False	80.08	United States	null
Law	Christopher2876	Student	7.55	Christopher	Male	Brown	True	32.45	United Kingdom	null
Law	Isabella3000	Student	18.07	Isabella	Female	Rosso	True	51.76	Italy	null
Law	Mia5426	Student	6.66	Mia	Female	Kim	False	30.99	South Korea	null
Law	Michelle9000	Student	13.06	Michelle	Female	Flores	True	77.52	Philippines	null
Law	Olivia0055	Student	32.91	Olivia	Female	Lee	False	60.07	France	null
Retail	Noah2247	Student	65.99	Noah	Male	Kim	False	7.01	United Kingdom	null
Retail	Rachel0376	Teacher	2.56	Rachel	Female	Adams	False	40.4	United States	null
Finance	Adewale3222	Student	4.82	Adewale	Female	Ayorinde	False	53.18	Nigeria	null
Finance	Evelyn2231	Teacher	9.66	Evelyn	Female	Rivera	True	42.84	Netherlands	null
Finance	Maddie5677	Student	4.95	Maddie	Female	Madison	False	28.85	United Kingdom	null
Finance	Mia1387	Student	125.32	Mia	Female	Yamamoto	True	25.99	Netherlands	null
Finance	Paola4552	Student	3.83	Paola	Female	Harris	False	66.21	France	null
Finance	Samantha1345	Student	2.22	Samantha	Female	Hernandez	False	177.77	Spain	null
Finance	Victoria9113	Student	9.77	Victoria	Female	Nelson	True	34.88	United Kingdom	null
Technology	Adekola0156	Student	23.21	Adekola	Male	Adeniyi	True	100.13	Nigeria	null
Technology	Amanda3333	Teacher	17.09	Amanda	Female	Martinez	False	50.81	Spain	null
Technology	Britanny5590	Teacher	1	Britanny	Female	Baker	False	69	United Kingdom	null
Technology	Eric8228	Student	7.98	Eric	Male	White	False	19.98	United Kingdom	null
Technology	Jessica3577	Teacher	7.11	Jessica	Female	Taylor	True	33.81	Spain	null
Technology	Kimberly6666	Teacher	0	Kimberly	Female	Rivera	False	235.66	Cuba	null
Technology	Logan3256	Teacher	15.99	Logan	Male	Wong	True	29.99	United Kingdom	null
Technology	Nicole1348	Student	13.54	Nicole	Female	Perez	True	67.09	Spain	null
Technology	Nicole5677	Student	22	Nicole	Female	Perez	True	56.89	Columbia	null
Technology	Timothy4552	Student	0.06	Timothy	Male	Wright	True	76.29	United States	null
Technology	Timothy5666	Student	7.77	Timothy	Male	Clark	False	12.22	United Kingdom	null
Technology	Tony4777	Student	1.04	Tony	Male	Garcia	True	38.96	United Kingdom	null
Technology	Trudy1111	Student	7.54	Trudy	Female	Jones	False	90.29	Netherlands	null
Manufacturing	Daniel6009	Student	77.54	Daniel	Male	Garcia	False	210.55	Spain	null
Manufacturing	Ferrero8080	Teacher	17.44	Ferrero	Male	Ferrari	False	50.42	Italy	null
Aviation	Anthony7684	Student	7.56	Anthony	Male	Montagnana	True	400.09	Italy	null
Aviation	Mersei2566	Student	67.22	Mersei	Female	Mongaba	True	51.27	United Kingdom	null
Aviation	Nicholas2784	Teacher	8.45	Nicholas	Male	Martinez	True	75.01	Mexico	null
Aviation	Victoria0057	Teacher	0.69	Victoria	Female	Nelson	False	87.21	United Kingdom	null

(35 rows)

This query gathers the ‘sum’ by industry, and is different to the previous code that took an the ‘avg’:

```
cqlsh:chippi> SELECT
          ... industry_of_choice,
          ... SUM(real_budget_spend) AS average_spend
          ... FROM
          ... chippi.duplicate_userbase
          ... GROUP BY industry_of_choice;

 industry_of_choice | average_spend
--------------------+---------------
                Law |        388.08
             Retail |         47.41
            Finance |        429.72
         Technology |        881.12
      Manufacturing |        260.97
           Aviation |        613.58

(6 rows)

Warnings :
Aggregation query used without partition key

cqlsh:chippi>
```

Addition of teacher_id into the main table created:

```
cqlsh:chippi> ALTER TABLE chippi.main_userbase ADD teacher_id text;
```

```
cqlsh:chippi> BEGIN BATCH
          ... UPDATE main_userbase SET teacher_id = 'Evelyn2231' WHERE user_id = 'Samantha1345';
          ... UPDATE main_userbase SET teacher_id = 'Evelyn2231' WHERE user_id = 'Paola4552';
          ... UPDATE main_userbase SET teacher_id = 'Jessica3577' WHERE user_id = 'Tony4777';
          ... UPDATE main_userbase SET teacher_id = 'Caitlin1355' WHERE user_id = 'Mia5426';
          ... UPDATE main_userbase SET teacher_id = 'Evelyn2231' WHERE user_id = 'Maddie5677';
          ... UPDATE main_userbase SET teacher_id = 'Evelyn2231' WHERE user_id = 'Mia1387';
          ... UPDATE main_userbase SET teacher_id = 'Evelyn2231' WHERE user_id = 'Victoria9113';
          ... UPDATE main_userbase SET teacher_id = 'Victoria4552' WHERE user_id = 'Mersei2566';
          ... UPDATE main_userbase SET teacher_id = 'Caitlin1355' WHERE user_id = 'Olivia0055';
          ... UPDATE main_userbase SET teacher_id = 'Evelyn2231' WHERE user_id = 'Adewale3222';
          ... UPDATE main_userbase SET teacher_id = 'Rachel0376' WHERE user_id = 'Noah2247';
          ... UPDATE main_userbase SET teacher_id = 'Logan3256' WHERE user_id = 'Eric8228';
          ... UPDATE main_userbase SET teacher_id = 'Caitlin1355' WHERE user_id = 'Isabella3000';
          ... UPDATE main_userbase SET teacher_id = 'Victoria0057' WHERE user_id = 'Anthony7684';
          ... UPDATE main_userbase SET teacher_id = 'Caitlin1355' WHERE user_id = 'Michelle9000';
          ... UPDATE main_userbase SET teacher_id = 'Britanny5590' WHERE user_id = 'Nicole1348';
          ... UPDATE main_userbase SET teacher_id = 'Ferrero8080' WHERE user_id = 'Daniel6009';
          ... UPDATE main_userbase SET teacher_id = 'Britanny5590' WHERE user_id = 'Nicole5677';
          ... UPDATE main_userbase SET teacher_id = 'Britanny5590' WHERE user_id = 'Trudy1111';
          ... UPDATE main_userbase SET teacher_id = 'Caitlin1355' WHERE user_id = 'Ava0234';
          ... UPDATE main_userbase SET teacher_id = 'Britanny5590' WHERE user_id = 'Timothy5666';
          ... UPDATE main_userbase SET teacher_id = 'Jessica3577' WHERE user_id = 'Adekola0156';
          ... UPDATE main_userbase SET teacher_id = 'Caitlin1355' WHERE user_id = 'Christopher2876';
          ... UPDATE main_userbase SET teacher_id = 'Britanny5590' WHERE user_id = 'Timothy4552';
          ... APPLY BATCH;
cqlsh:chippi>
```

This query selects students that share the same Finance tutor 'Evelyn'.

```
cqlsh:chippi> SELECT user_id, industry_of_choice
          ... FROM chippi.main_userbase
          ... WHERE teacher_id = 'Evelyn2231' ALLOW FILTERING;

 user_id      | industry_of_choice
--------------+--------------------
 Samantha1345 |            Finance
    Paola4552 |            Finance
  Maddie5677 |            Finance
     Mia1387 |            Finance
 Victoria9113 |            Finance
  Adewale3222 |            Finance

(6 rows)
cqlsh:chippi>
```

This query selects students according to their teacher ID and details the commonality in industry:

```
cqlsh:chippi> SELECT teacher_id, user_id, industry_of_choice
          ... FROM chippi.main_userbase
          ... WHERE account_type = 'Student' ALLOW FILTERING;

 teacher_id   | user_id         | industry_of_choice
--------------+-----------------+--------------------
   Evelyn2231 |    Samantha1345 |            Finance
   Evelyn2231 |       Paola4552 |            Finance
  Jessica3577 |        Tony4777 |         Technology
  Caitlin1355 |         Mia5426 |                Law
   Evelyn2231 |      Maddie5677 |            Finance
   Evelyn2231 |         Mia1387 |            Finance
   Evelyn2231 |    Victoria9113 |            Finance
 Victoria4552 |      Mersei2566 |           Aviation
  Caitlin1355 |      Olivia0055 |                Law
   Evelyn2231 |     Adewale3222 |            Finance
   Rachel0376 |        Noah2247 |             Retail
    Logan3256 |        Eric8228 |         Technology
  Caitlin1355 |    Isabella3000 |                Law
 Victoria0057 |     Anthony7684 |           Aviation
  Caitlin1355 |    Michelle9000 |                Law
 Britanny5590 |      Nicole1348 |         Technology
  Ferrero8080 |      Daniel6009 |      Manufacturing
 Britanny5590 |      Nicole5677 |         Technology
 Britanny5590 |       Trudy1111 |         Technology
  Caitlin1355 |         Ava0234 |                Law
 Britanny5590 |     Timothy5666 |         Technology
  Jessica3577 |     Adekola0156 |         Technology
  Caitlin1355 | Christopher2876 |                Law
 Britanny5590 |     Timothy4552 |         Technology

(24 rows)
cqlsh:chippi>
```

Altering table to include a column that records the last log in for each user:

```
cqlsh:chippi> ALTER TABLE chippi.main_userbase
          ... ADD last_log_in timestamp;
```

Inputting data:

```
cqlsh:chippi> UPDATE main_userbase
          ... SET last_log_in = '2024-02-20T12:00:00Z'
          ... WHERE user_id = 'Caitlin1355';
cqlsh:chippi>
```

```
cqlsh:chippi> BEGIN BATCH
          ... UPDATE main_userbase SET last_log_in = '2024-01-13T09:58:13Z' WHERE user_id = 'Christopher2876';
          ... UPDATE main_userbase SET last_log_in = '2024-02-16T13:51:50Z' WHERE user_id = 'Isabella3000';
          ... UPDATE main_userbase SET last_log_in = '2024-01-23T10:00:41Z' WHERE user_id = 'Mia5426';
          ... UPDATE main_userbase SET last_log_in = '2024-02-01T11:05:32Z' WHERE user_id = 'Michelle9000';
          ... UPDATE main_userbase SET last_log_in = '2024-01-20T16:20:20Z' WHERE user_id = 'Olivia0055';
          ... UPDATE main_userbase SET last_log_in = '2023-02-21T08:43:20Z' WHERE user_id = 'Michelle9000';
          ... UPDATE main_userbase SET last_log_in = '2023-12-22T22:01:53Z' WHERE user_id = 'Mersei2566';
          ... UPDATE main_userbase SET last_log_in = '2024-02-19T14:03:23Z' WHERE user_id = 'Olivia0055';
          ... UPDATE main_userbase SET last_log_in = '2024-01-23T10:00:41Z' WHERE user_id = 'Noah2247';
          ... UPDATE main_userbase SET last_log_in = '2024-01-01T09:45:22Z' WHERE user_id = 'Rachel0376';
          ... UPDATE main_userbase SET last_log_in = '2023-11-20T13:22:54Z' WHERE user_id = 'Adewale3222';
          ... UPDATE main_userbase SET last_log_in = '2022-02-22T15:30:31Z' WHERE user_id = 'Evelyn2231';
          ... UPDATE main_userbase SET last_log_in = '2024-01-04T05:43:58Z' WHERE user_id = 'Maddie5677';
          ... UPDATE main_userbase SET last_log_in = '2024-01-12T19:31:50Z' WHERE user_id = 'Mia1387';
          ... UPDATE main_userbase SET last_log_in = '2023-11-11T13:01:22Z' WHERE user_id = 'Paola4552';
          ... UPDATE main_userbase SET last_log_in = '2024-02-20T10:18:11Z' WHERE user_id = 'Samantha1345';
          ... UPDATE main_userbase SET last_log_in = '2024-01-17T10:18:11Z' WHERE user_id = 'Victoria9113';
          ... UPDATE main_userbase SET last_log_in = '2020-10-31T10:18:11Z' WHERE user_id = 'Adekola0156';
          ... UPDATE main_userbase SET last_log_in = '2024-03-11T10:18:11Z' WHERE user_id = 'Amanda3333';
          ... UPDATE main_userbase SET last_log_in = '2023-04-30T10:18:11Z' WHERE user_id = 'Britanny5590';
          ... UPDATE main_userbase SET last_log_in = '2022-09-27T10:18:11Z' WHERE user_id = 'Eric8228';
          ... UPDATE main_userbase SET last_log_in = '2023-07-23T10:18:11Z' WHERE user_id = 'Jessica3577';
          ... UPDATE main_userbase SET last_log_in = '2024-02-08T10:18:11Z' WHERE user_id = 'Kimberly6666';
          ... UPDATE main_userbase SET last_log_in = '2024-02-06T10:18:11Z' WHERE user_id = 'Logan3256';
          ... UPDATE main_userbase SET last_log_in = '2024-01-18T10:18:11Z' WHERE user_id = 'Nicole1348';
          ... UPDATE main_userbase SET last_log_in = '2020-11-29T10:18:11Z' WHERE user_id = 'Nicole5677';
          ... UPDATE main_userbase SET last_log_in = '2020-09-24T10:18:11Z' WHERE user_id = 'Timothy4552';
          ... UPDATE main_userbase SET last_log_in = '2021-10-14T10:18:11Z' WHERE user_id = 'Timothy5666';
          ... UPDATE main_userbase SET last_log_in = '2022-12-22T10:18:11Z' WHERE user_id = 'Tony4777';
          ... UPDATE main_userbase SET last_log_in = '2021-10-05T10:18:11Z' WHERE user_id = 'Trudy1111';
          ... UPDATE main_userbase SET last_log_in = '2023-02-03T10:18:11Z' WHERE user_id = 'Daniel6009';
          ... UPDATE main_userbase SET last_log_in = '2023-01-21T10:18:11Z' WHERE user_id = 'Ferrero8080';
          ... UPDATE main_userbase SET last_log_in = '2024-01-16T10:18:11Z' WHERE user_id = 'Anthony7684';
          ... UPDATE main_userbase SET last_log_in = '2024-01-04T10:18:11Z' WHERE user_id = 'Mersei2566';
          ... UPDATE main_userbase SET last_log_in = '2024-02-07T10:18:11Z' WHERE user_id = 'Nicholas2784';
          ... UPDATE main_userbase SET last_log_in = '2024-02-13T10:18:11Z' WHERE user_id = 'Victoria0057';
          ... APPLY BATCH;
cqlsh:chippi>
```

The following query filters all the ‘last log in’ timestamps of people from before the date 2021-06-13, which could be useful for marketing to idle users:

```
cqlsh:chippi> SELECT user_id, last_log_in
          ... FROM chippi.main_userbase
          ... WHERE last_log_in < toTimestamp('2021-06-13') ALLOW FILTERING;

 user_id     | last_log_in
-------------+---------------------------------
  Nicole5677 | 2020-11-29 10:18:11.000000+0000
 Adekola0156 | 2020-10-31 10:18:11.000000+0000
 Timothy4552 | 2020-09-24 10:18:11.000000+0000

(3 rows)
cqlsh:chippi>
```

This query acts as a log that allows for analysis of demographics to provide personalised marketing/recommendations to users:

```
cqlsh:chippi> SELECT user_id, first_name, last_name, gender, region
          ... FROM chippi.main_userbase;

 user_id         | first_name  | last_name   | gender | region
-----------------+-------------+-------------+--------+----------------
    Britanny5590 |    Britanny |      Baker | Female | United Kingdom
     Ferrero8080 |     Ferrero |     Ferrari |   Male |          Italy
   Samantha1345 |    Samantha |   Hernandez | Female |          Spain
      Paola4552 |       Paola |      Harris | Female |         France
       Tony4777 |        Tony |      Garcia |   Male | United Kingdom
        Mia5426 |         Mia |         Kim | Female |    South Korea
     Maddie5677 |      Maddie |     Madison | Female | United Kingdom
        Mia1387 |         Mia |    Yamamoto | Female |    Netherlands
   Nicholas2784 |    Nicholas |    Martinez |   Male |         Mexico
   Kimberly6666 |    Kimberly |      Rivera | Female |           Cuba
   Victoria9113 |    Victoria |      Nelson | Female | United Kingdom
     Mersei2566 |      Mersei |     Mongaba | Female | United Kingdom
     Olivia0055 |      Olivia |         Lee | Female |         France
    Jessica3577 |     Jessica |      Taylor | Female |          Spain
    Adewale3222 |     Adewale |    Ayorinde | Female |        Nigeria
       Noah2247 |        Noah |         Kim |   Male | United Kingdom
     Evelyn2231 |      Evelyn |      Rivera | Female |    Netherlands
      Eric8228 |       Eric |       White |   Male | United Kingdom
   Isabella3000 |    Isabella |       Rosso | Female |          Italy
    Anthony7684 |     Anthony |  Montagnana |   Male |          Italy
     Rachel0376 |      Rachel |       Adams | Female |  United States
   Michelle9000 |    Michelle |      Flores | Female |    Philippines
     Nicole1348 |      Nicole |       Perez | Female |          Spain
     Daniel6009 |      Daniel |      Garcia |   Male |          Spain
     Nicole5677 |      Nicole |       Perez | Female |       Columbia
    Caitlin1355 |     Caitlin |       Nunez | Female |  United States
      Trudy1111 |       Trudy |       Jones | Female |    Netherlands
       Ava0234 |         Ava |    Sullivan | Female |  United States
    Timothy5666 |     Timothy |       Clark |   Male | United Kingdom
    Adekola0156 |     Adekola |     Adeniyi |   Male |        Nigeria
     Amanda3333 |      Amanda |    Martinez | Female |          Spain
 Christopher2876 | Christopher |       Brown |   Male | United Kingdom
      Logan3256 |       Logan |        Wong |   Male | United Kingdom
   Victoria0057 |    Victoria |      Nelson | Female | United Kingdom
    Timothy4552 |     Timothy |      Wright |   Male |  United States

(35 rows)
```

2. MongoDB

The concept for the ‘bonnebouche database is a directory and digital marketing platform for chefs and stores associated with confectionaries and desserts. Users can query the database to filter through information related to the dessert products and stores for specific needs in search. An advantage of using MongoDB is the querying ability. The database allows for complex queries that can allow users to find specific information efficiently. A disadvantage is that the lack of schema validation could create inconsistencies in data, such as mixed entries of an entity. I have provided screenshots and comments below according to the commands and queries displayed:

Creation of database:

```
ususe bonnebouche
switched to db bonnebouche
> db
bonnebouche
>
```

Creation of cinnamonrolls document which represents the attributes that make up the dessert:

```
> cinnamonrolls = {_id: "CRPLP128",
... Title: "Cinnamon Rolls",
... DessertType: "Pastry",
... HeadChef: "Layli Patron",
... Ingredients: "300g self raising flour, 4 tsp caster sugar, 2 tsp ground cinnamon, 110g butter, 2 egg, 130ml milk, icin
g",
... Instructions: "Mix filling ingredients together. Spread evenly over dough then roll up like a Swiss roll. Cut dough in
to slices and pack into prepared tin. Brush gently with milk and bake for approximately 30 mins. Remove from oven and cool
 for 5 mins before removing tin. Drizzle icing over the rolls"
... }
{
        "_id" : "CRPLP128",
        "Title" : "Cinnamon Rolls",
        "DessertType" : "Pastry",
        "HeadChef" : "Layli Patron",
        "Ingredients" : "300g self raising flour, 4 tsp caster sugar, 2 tsp ground cinnamon, 110g butter, 2 egg, 130ml mil
k, icing",
        "Instructions" : "Mix filling ingredients together. Spread evenly over dough then roll up like a Swiss roll. Cut d
ough into slices and pack into prepared tin. Brush gently with milk and bake for approximately 30 mins. Remove from oven a
nd cool for 5 mins before removing tin. Drizzle icing over the rolls"
}
```

Insertion of “cinnamonrolls” into a collection that has been called ‘desserts_portfolio’ (portfolio of desserts and stores that sell similar products). I ran a query within the collection to confirm that insertion of the document within:

```
> db.desserts_portfolio.insert(cinnamonrolls)
WriteResult({ "nInserted" : 1 })
> db.desserts_portfolio.find()
{ "_id" : "CRPLP128", "Title" : "Cinnamon Rolls", "DessertType" : "Pastry", "HeadChef" : "Layli Patron", "Ingredients" : "
300g self raising flour, 4 tsp caster sugar, 2 tsp ground cinnamon, 110g butter, 2 egg, 130ml milk, icing", "Instructions"
 : "Mix filling ingredients together. Spread evenly over dough then roll up like a Swiss roll. Cut dough into slices and p
ack into prepared tin. Brush gently with milk and bake for approximately 30 mins. Remove from oven and cool for 5 mins bef
ore removing tin. Drizzle icing over the rolls" }
```

Confirmation of the “desserts_portfolio” collection:

```
> show collections
desserts_portfolio
```

The code seen above acts as a standard ‘structure’ for some of the following entries

```
> honeybaklava = {_id: "HBPLA001",
... Title: "Honey Baklava",
... DessertType: "Pastry",
... HeadChef: "Leila Al-Farsi",
... Ingredients: "Phyllo dough, pistachoi nuts, butter, cinnamon, honey syrup",
... Instructions: "Stack sheets of phyllo dough, glaze each layer with butter, sprinkle nuts
 and cinnamon, repeat until layer formed. Cut baklava into shapes, bake until golden crisp.
Remove from oven and pour syrup over it."
... }
{
        "_id" : "HBPLA001",
        "Title" : "Honey Baklava",
        "DessertType" : "Pastry",
        "HeadChef" : "Leila Al-Farsi",
        "Ingredients" : "Phyllo dough, pistachoi nuts, butter, cinnamon, honey syrup",
        "Instructions" : "Stack sheets of phyllo dough, glaze each layer with butter, sprink
le nuts and cinnamon, repeat until layer formed. Cut baklava into shapes, bake until golden
crisp. Remove from oven and pour syrup over it."
}
> db.desserts_portfolio.insert(honeybaklava)
WriteResult({ "nInserted" : 1 })
```

```
> chocolatechipcookies = {_id: "CCCBS003",
... Title: "Chocolate Chip Cookies",
... DessertType: "Biscuit",
... HeadChef: "Sophie Bennett",
... Ingredients: "120g butter, 75g light brown sugar, 75 golden caster sugar, 1 egg, 1 tsp vanilla extract,
180g plain flour, 1/2 tsp bicarbonate of soda, 150g dark chocolate",
... Instructions: "Whisk together sugars, salt and butter until paste forms. Whisk in egg and vanilla. Add i
n flour and baking soda, then fold the mixture. Fold in the chocolate chunks, then chill dough for 30 mins.
Preheat oven to 180C, line a baking sheet with parchment paper. Scoop numbers of the dough onto baking sheet
 and bake for approximately 15 mins."
... }
{
        "_id" : "CCCBS003",
        "Title" : "Chocolate Chip Cookies",
        "DessertType" : "Biscuit",
        "HeadChef" : "Sophie Bennett",
        "Ingredients" : "120g butter, 75g light brown sugar, 75 golden caster sugar, 1 egg, 1 tsp vanilla ex
tract, 180g plain flour, 1/2 tsp bicarbonate of soda, 150g dark chocolate",
        "Instructions" : "Whisk together sugars, salt and butter until paste forms. Whisk in egg and vanilla
. Add in flour and baking soda, then fold the mixture. Fold in the chocolate chunks, then chill dough for 30
 mins. Preheat oven to 180C, line a baking sheet with parchment paper. Scoop numbers of the dough onto bakin
g sheet and bake for approximately 15 mins."
}
```

```
> db.desserts_portfolio.insert(chocolatechipcookies)
WriteResult({ "nInserted" : 1 })
```

The ‘bonnebouche’ database in the list of databases (show dbs) which indicates it's no longer empty:

```
> show collections
desserts_portfolio
> show dbs
admin        0.000GB
bonnebouche  0.000GB
config       0.000GB
local        0.000GB
movies       0.000GB
>
```

```
> brownies = {_id: "BCO2002",
... Title: "Brownies",
... DessertType: "Cake",
... HeadChef: "Oliver Smith",
... Ingredients: "2 egg, water, powdered sugar, unsweetened cocoa powder, oil, 1/2 tsp vanilla extract",
... Instructions: "Mix together dry and wet ingredients in two seperate bowls. Sprinkle dry mixture over wet
 one and fold until combined. Pour the batter into a baking pan lined with parchment paper. Bake pan within
a 165C oven and bake for approximately minutes."
... }
{
        "_id" : "BCO2002",
        "Title" : "Brownies",
        "DessertType" : "Cake",
        "HeadChef" : "Oliver Smith",
        "Ingredients" : "2 egg, water, powdered sugar, unsweetened cocoa powder, oil, 1/2 tsp vanilla extrac
t",
        "Instructions" : "Mix together dry and wet ingredients in two seperate bowls. Sprinkle dry mixture o
ver wet one and fold until combined. Pour the batter into a baking pan lined with parchment paper. Bake pan
within a 165C oven and bake for approximately minutes."
}
> db.desserts_portfolio.insert(brownies)
WriteResult({ "nInserted" : 1 })
```

```
> vanillacupcake = {_id: "VCCAP980",
... Title: "Vanilla Cupcakes",
... DessertType: "Cake",
... HeadChef: "Adam Pearson",
... Ingredients: "120g butter, 120g caster sugar, 2 egg, 1 tsp vanilla extract, 120g self raising flour, but
tercream icing",
... Instructions: "Heat oven to 180C and line a 12-hole muffin tin with paper cases. Cream the butter and su
gar together in bowl. Beat eggs in a separate bowl and mix into butter mixture alojg with vanilla extract. F
old in the flour, add a little milk for good consistency. Spoon the mixture into the paper cases until they
are three quarters full. Bake in the oven for 15 mins, remove then set aside to cool for 5-10 mins."
... }
{
        "_id" : "VCCAP980",
        "Title" : "Vanilla Cupcakes",
        "DessertType" : "Cake",
        "HeadChef" : "Adam Pearson",
        "Ingredients" : "120g butter, 120g caster sugar, 2 egg, 1 tsp vanilla extract, 120g self raising flo
ur, buttercream icing",
        "Instructions" : "Heat oven to 180C and line a 12-hole muffin tin with paper cases. Cream the butter
 and sugar together in bowl. Beat eggs in a separate bowl and mix into butter mixture alojg with vanilla ext
ract. Fold in the flour, add a little milk for good consistency. Spoon the mixture into the paper cases unti
l they are three quarters full. Bake in the oven for 15 mins, remove then set aside to cool for 5-10 mins."
}
> db.desserts_portfolio.insert(vanillacupcake)
WriteResult({ "nInserted" : 1 })
```

```
> applepie = {_id: "APPM005",
... Title: "Apple Pie",
... DessertType: "Pastry",
... HeadChef: "Sophie Bennett"
... }
{
        "_id" : "APPM005",
        "Title" : "Apple Pie",
        "DessertType" : "Pastry",
        "HeadChef" : "Sophie Bennett"
}
> db.desserts_portfolio.insert(applepie)
WriteResult({ "nInserted" : 1 })
```

```
> cheesecake = {_id: "CCJC200",
... Title: "Cheesecake",
... DessertType: "Cake",
... HeadChef: "Jessica Carter"
... }
{
        "_id" : "CCJC200",
        "Title" : "Cheesecake",
        "DessertType" : "Cake",
        "HeadChef" : "Jessica Carter"
}
> db.desserts_portfolio.insert(cheesecake)
WriteResult({ "nInserted" : 1 })
```

```
> redvelvet = {_id: "RVCCB050",
... Title: "Red Velvet Cake",
... DessertType: "Cake",
... HeadChef: "Cassie Best",
... Ingredients: "300ml vegetable oil, 500g plain flour, 2 tsp cocoa powder, 4 tsp baking powder, 2 tsp bica
rbonate soda, 560g light brown soft sugar, 1 tsp fine salt, 400ml buttermilk, 4 tsp vanilla extract, 30ml re
d floor colouring, 4 egg, icing",
... Instructions: "Heat oven to 180C. Oil and line the base and sides of two 20cm cake tins with baking parc
hment. Put half each of the flour, cocoa powder, baking powder, bicarb, sugar and salt in a bowl and mix wel
l. Mix half each of the buttermilk, oil, vanilla extract, food colouring and 100ml water in jug. Add 2 eggs
and whisk until smooth. Pour wet ingredients into the dry and whisk until well combined. Pour the cake mixtu
re evenly into two tins and bake for approximately 30 mins. Cool for 10 mins, then add icing."
... }
{
        "_id" : "RVCCB050",
        "Title" : "Red Velvet Cake",
        "DessertType" : "Cake",
        "HeadChef" : "Cassie Best",
        "Ingredients" : "300ml vegetable oil, 500g plain flour, 2 tsp cocoa powder, 4 tsp baking powder, 2 t
sp bicarbonate soda, 560g light brown soft sugar, 1 tsp fine salt, 400ml buttermilk, 4 tsp vanilla extract,
30ml red floor colouring, 4 egg, icing",
        "Instructions" : "Heat oven to 180C. Oil and line the base and sides of two 20cm cake tins with baki
ng parchment. Put half each of the flour, cocoa powder, baking powder, bicarb, sugar and salt in a bowl and
mix well. Mix half each of the buttermilk, oil, vanilla extract, food colouring and 100ml water in jug. Add
2 eggs and whisk until smooth. Pour wet ingredients into the dry and whisk until well combined. Pour the cak
e mixture evenly into two tins and bake for approximately 30 mins. Cool for 10 mins, then add icing."
}
> db.desserts_portoflio.insert(redvelvet)
WriteResult({ "nInserted" : 1 })
```

```
> milkshake = {_id: "MCRN051",
... Title: "Milkshake",
... DessertType: "Cake",
... HeadChef: "Raj Nicholas"
... }
{
        "_id" : "MCRN051",
        "Title" : "Milkshake",
        "DessertType" : "Cake",
        "HeadChef" : "Raj Nicholas"
}
> db.desserts_portfolio.insert(milkshake)
WriteResult({ "nInserted" : 1 })
>
```

Batch insertion of dessert stores:

```
> db.desserts_portfolio.insertMany([
... {
... _id: "GAILS007",
... StoreName: "GAIL's",
... HeadChef: "Cassie Best",
... Address: ["24 - 25 Nelson Rd, Greenwich, London", "SE10 9JB"],
... AvailabilityTime: "6:30am - 8pm",
... YearOfEStablishment: "2006",
... Capacity: "110",
... ProductsSold: "Brownies, Chocolate Muffins, Cinnamon Rolls, Crossaints"
... },
... {
... _id: "PARKS800",
... StoreName: "Parkside Cafe",
... HeadChef: "Oliver Smith",
... Address: ["23 Major Draper St, Royal Arsenal, London", "SE18 6ZF"],
... AvailabilityTime: "7am - 9pm",
... YearOfEstablishment: "1999",
... Capacity: "110",
... ProductsSold: "Brownies, Chocolate Muffins, Cinnamon Rolls, Crossaints"
... }])
{ "acknowledged" : true, "insertedIds" : [ "GAILS007", "PARKS800" ] }
>
```

I originally duplicated the entry for the ‘ProductsSold’ which is meant to differ for GAILS but updated this with the following command:

```
> db.desserts_portfolio.update({_id: "GAILS007"},
... {$set:{ProductsSold: "Red Velvet Cake, Crossaints, Chocolate Chip Cupcakes, Brownies"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

```
> db.desserts_portfolio.insertMany([
... {
...   _id: "HEAVE422",
...   StoreName: "Heavenly Desserts",
...   HeadChef: "Leila Al-Farsi",
...   Address: ["3 - 4 Endeavour Square, Stratford, London", "E15 1JN"],
...   AvailabilityTime: "12pm - 11pm",
...   YearOfEstablishment: "2020",
...   Capacity: "180",
...   ProductsSold: "Baklava, Donuts, Mazurek, Cinnamon Rolls"
... },
... {
...   _id: "ZBAR574",
...   StoreName: "Z Bar",
...   HeadChef: "Michael Thompson",
...   Address: ["58 Stoke Newington High St, Stoke Newington, London", "N16 7PB"],
...   AvailabilityTime: "7am - 6:30pm",
...   YearOfEstablishment: "2013",
...   Capacity: "100",
...   ProductsSold: "Apple Pie, Chocolate Cakes, Crepes, Waffles"
... },
... {
...   _id: "BEIG002",
...   StoreName: "Beigel Bake",
...   HeadChef: "Adam Pearson",
...   Address: ["159 Brick Lane, Whitechapel, London", "E1 6SB"],
...   AvailabilityTime: "Open 24 hours",
...   YearOfEstablishment: "2011",
...   Capacity: "45",
...   ProductsSold: "Croissants, Vanilla Cupcakes, Chocolate Waffles, Filled Bagels"
... },
... {
...   _id: "THEB001",
...   StoreName: "The Bakery",
...   HeadChef: "Layli Patron",
...   Address: ["6 Bexley High St, Bexley, London", "DA5 1AD"],
...   AvailabilityTime: "7am - 4pm",
...   YearOfEstablishment: "2010",
...   Capacity: "120",
...   ProductsSold: "Cinnamon Rolls, Pancakes, Muffins, Cakes"
... },
... {
...   _id: "THEH223",
...   StoreName: "The Hummingbird",
...   HeadChef: "Jessica Carter",
...   Address: ["47 Old Brompton Rd, South Kensington, London", "SW7 3JP"],
...   AvailabilityTime: "8am - 7pm",
...   YearOfEstablishment: "2018",
...   Capacity: "92",
...   ProductsSold: "Cheesecake, Donuts, Cupcakes, Coffee"
... }])
{
        "acknowledged" : true,
        "insertedIds" : [
                "HEAVE422",
                "ZBAR574",
                "BEIG002",
                "THEB001",
                "THEH223"
        ]
}
>
```

```
> db.desserts_portfolio.insertMany([
...   {
...     _id: "CAKEB019",
...     StoreName: "Cake Box",
...     HeadChef: "Charlotte Baker",
...     Address: ["333 Ballards Ln, North Finchley, London", "N12 8LT"],
...     AvailabilityTime: "10am - 7pm",
...     YearOfEstablishment: "2015",
...     Capacity: "50",
...     ProductsSold: "Vanilla Cakes, Chocolate Muffins, Crepes, Ice cream"
...   },
...   {
...     _id: "FARM088",
...     StoreName: "Farm Girl",
...     HeadChef: "Sophie McDermont",
...     Address: ["59A Portobello Rd, Notting Hill, London", "W11 3DB"],
...     AvailabilityTime: "9am - 5:30pm",
...     YearOfEstablishment: "2005",
...     Capacity: "85",
...     ProductsSold: "Bagels, Croissants, Donuts, Chocolate Cupcakes"
...   },
...   {
...     _id: "OLDS067",
...     StoreName: "Old Spike Peckham",
...     HeadChef: "Emma Wilson",
...     Address: ["54 Peckham Rye, Peckham, London", "SE15 4JR"],
...     AvailabilityTime: "9am - 5pm",
...     YearOfEstablishment: "2019",
...     Capacity: "150",
...     ProductsSold: "Cheesecake, Vanilla Cupcakes, Burgers, Wings, Coffee, Alcohol"
...   },
...   {
...     _id: "DONO689",
...     StoreName: "Donovan’s bakehouse",
...     HeadChef: "Sophie Kim",
...     Address: ["Old Spitalfields Market, Aldgate, London", "E1 6QS"],
...     AvailabilityTime: "9:30am - 5pm",
...     YearOfEstablishment: "2017",
...     Capacity: "80",
...     ProductsSold: "Bagels, Bread, Chocolate Chip Cookies, Brownies"
...   },
...   {
...     _id: "CAFE802",
...     StoreName: "Cafe East",
...     HeadChef: "Daniel Rodriguez",
...     Address: ["426 Roman Rd, Bow, London", "E3 5LU"],
...     AvailabilityTime: "8:30am - 5pm",
...     YearOfEstablishment: "2019",
...     Capacity: "100",
...     ProductsSold: "Full English Breakfast, Pancakes, Coffee, Tea, Donuts"
...   },
...   {
...     _id: "CASH418",
...     StoreName: "Cakes & Shakes",
...     HeadChef: "Raj Nicholas",
...     Address: ["587 Cranbook Rd, Gants Hill, Ilford", "IG2 6JZ"],
...     AvailabilityTime: "10am - 11pm",
...     YearOfEstablishment: "2010",
...     Capacity: "300",
...     ProductsSold: "Crepes, Waffles, Milkshake, Ice cream"
...   },
...   {
...     _id: "AOJP302",
...     StoreName: "Afters Orginal",
...     HeadChef: "James Paterson",
...     Address: ["166 Green St, Forest Gate, London", "E7 8JT"],
...     AvailabilityTime: "3pm - 1am",
...     YearOfEstablishment: "2009",
...     Capacity: "200",
...     ProductsSold: "Coffee, Red Velvet Cake, Crepes, Waffles"
...   }
... ])
{
        "acknowledged" : true,
        "insertedIds" : [
                "CAKEB019",
                "FARM088",
                "OLDS067",
                "DONO689",
                "CAFE802",
                "CASH418",
                "AOJP302"
        ]
}
```

The following command showcases all documents that exist within the collection ‘desserts_portfolio’:

```
> db.desserts_portfolio.find().pretty()
{
        "_id" : "CRPLP128",
        "Title" : "Cinnamon Rolls",
        "DessertType" : "Pastry",
        "HeadChef" : "Layli Patron",
        "Ingredients" : "300g self raising flour, 4 tsp caster sugar, 2 tsp ground cinnamon, 110g butter
, 2 egg, 130ml milk, icing",
        "Instructions" : "Mix filling ingredients together. Spread evenly over dough then roll up like a
 Swiss roll. Cut dough into slices and pack into prepared tin. Brush gently with milk and bake for appro
ximately 30 mins. Remove from oven and cool for 5 mins before removing tin. Drizzle icing over the rolls
"
}
{
        "_id" : "HBPLA001",
        "Title" : "Honey Baklava",
        "DessertType" : "Pastry",
        "HeadChef" : "Leila Al-Farsi",
        "Ingredients" : "Phyllo dough, pistachoi nuts, butter, cinnamon, honey syrup",
        "Instructions" : "Stack sheets of phyllo dough, glaze each layer with butter, sprinkle nuts and
cinnamon, repeat until layer formed. Cut baklava into shapes, bake until golden crisp. Remove from oven
and pour syrup over it."
}
{
        "_id" : "CCCBS003",
        "Title" : "Chocolate Chip Cookies",
        "DessertType" : "Biscuit",
        "HeadChef" : "Sophie Bennett",
        "Ingredients" : "120g butter, 75g light brown sugar, 75 golden caster sugar, 1 egg, 1 tsp vanill
a extract, 180g plain flour, 1/2 tsp bicarbonate of soda, 150g dark chocolate",
        "Instructions" : "Whisk together sugars, salt and butter until paste forms. Whisk in egg and van
illa. Add in flour and baking soda, then fold the mixture. Fold in the chocolate chunks, then chill doug
h for 30 mins. Preheat oven to 180C, line a baking sheet with parchment paper. Scoop numbers of the doug
h onto baking sheet and bake for approximately 15 mins."
}
{
        "_id" : "BCO2002",
        "Title" : "Brownies",
        "DessertType" : "Cake",
        "HeadChef" : "Oliver Smith",
        "Ingredients" : "2 egg, water, powdered sugar, unsweetened cocoa powder, oil, 1/2 tsp vanilla ex
tract",
        "Instructions" : "Mix together dry and wet ingredients in two seperate bowls. Sprinkle dry mixtu
re over wet one and fold until combined. Pour the batter into a baking pan lined with parchment paper. B
ake pan within a 165C oven and bake for approximately minutes."
```

Note the accidental entry ‘YearOfEStablishment’ and duplicate ‘ProductsSOld’ for _id: “GAILS007” which will later be amended.

```
}
{
        "_id" : "APPM005",
        "Title" : "Apple Pie",
        "DessertType" : "Pastry",
        "HeadChef" : "Sophie Bennett"
}
{
        "_id" : "CCJC200",
        "Title" : "Cheesecake",
        "DessertType" : "Cake",
        "HeadChef" : "Jessica Carter"
}
{
        "_id" : "VCCAP980",
        "Title" : "Vanilla Cupcakes",
        "DessertType" : "Cake",
        "HeadChef" : "Adam Pearson",
        "Ingredients" : "120g butter, 120g caster sugar, 2 egg, 1 tsp vanilla extract, 120g s
 flour, buttercream icing",
        "Instructions" : "Heat oven to 180C and line a 12-hole muffin tin with paper cases. C
tter and sugar together in bowl. Beat eggs in a separate bowl and mix into butter mixture alo
illa extract. Fold in the flour, add a little milk for good consistency. Spoon the mixture in
r cases until they are three quarters full. Bake in the oven for 15 mins, remove then set asi
for 5-10 mins."
}
{
        "_id" : "MCRN051",
        "Title" : "Milkshake",
        "DessertType" : "Cake",
        "HeadChef" : "Raj Nicholas"
}
{
        "_id" : "GAILS007",
        "StoreName" : "GAIL's",
        "HeadChef" : "Cassie Best",
        "Address" : [
                "24 - 25 Nelson Rd, Greenwich, London",
                "SE10 9JB"
        ],
        "AvailabilityTime" : "6:30am - 8pm",
        "YearOfEStablishment" : "2006",
        "Capacity" : "110",
        "ProductsSold" : "Red Velvet Cake, Crossaints, Chocolate Chip Cupcakes, Brownies",
        "ProductsSOld" : "Red Velvet CAke, Crossaints, Chocolate Chip Cupcakes, Brownies"
}
```

```
{
        "_id" : "PARKS800",
        "StoreName" : "Parkside Cafe",
        "HeadChef" : "Oliver Smith",
        "Address" : [
                "23 Major Draper St, Royal Arsenal, London",
                "SE18 6ZF"
        ],
        "AvailabilityTime" : "7am - 9pm",
        "YearOfEstablishment" : "1999",
        "Capacity" : "110",
        "ProductsSold" : "Brownies, Chocolate Muffins, Cinnamon Rolls, Crossaints"
}
{
        "_id" : "HEAVE422",
        "StoreName" : "Heavenly Desserts",
        "HeadChef" : "Leila Al-Farsi",
        "Address" : [
                "3 - 4 Endeavour Square, Stratford, London",
                "E15 1JN"
        ],
        "AvailabilityTime" : "12pm - 11pm",
        "YearOfEstablishment" : "2020",
        "Capacity" : "180",
        "ProductsSold" : "Baklava, Donuts, Mazurek, Cinnamon Rolls"
}
{
        "_id" : "ZBAR574",
        "StoreName" : "Z Bar",
        "HeadChef" : "Michael Thompson",
        "Address" : [
                "58 Stoke Newington High St, Stoke Newington, London",
                "N16 7PB"
        ],
        "AvailabilityTime" : "7am - 6:30pm",
        "YearOfEstablishment" : "2013",
        "Capacity" : "100",
        "ProductsSold" : "Apple Pie, Chocolate Cakes, Crepes, Waffles"
}
{
        "_id" : "BEIG002",
        "StoreName" : "Beigel Bake",
        "HeadChef" : "Adam Pearson",
        "Address" : [
                "159 Brick Lane, Whitechapel, London",
                "E1 6SB"
        ],
        "AvailabilityTime" : "Open 24 hours",
        "YearOfEstablishment" : "2011",
        "Capacity" : "45",
        "ProductsSold" : "Croissants, Vanilla Cupcakes, Chocolate Waffles, Filled Bagels"
}
{
        "_id" : "THEB001",
        "StoreName" : "The Bakery",
        "HeadChef" : "Layli Patron",
        "Address" : [
                "6 Bexley High St, Bexley, London",
                "DA5 1AD"
        ],
        "AvailabilityTime" : "7am - 4pm",
        "YearOfEstablishment" : "2010",
        "Capacity" : "120",
        "ProductsSold" : "Cinnamon Rolls, Pancakes, Muffins, Cakes"
}
{
        "_id" : "THEH223",
        "StoreName" : "The Hummingbird",
        "HeadChef" : "Jessica Carter",
        "Address" : [
                "47 Old Brompton Rd, South Kensington, London",
                "SW7 3JP"
        ],
        "AvailabilityTime" : "8am - 7pm",
        "YearOfEstablishment" : "2018",
        "Capacity" : "92",
        "ProductsSold" : "Cheesecake, Donuts, Cupcakes, Coffee"
}
{
        "_id" : "CAKEB019",
        "StoreName" : "Cake Box",
        "HeadChef" : "Charlotte Baker",
        "Address" : [
                "333 Ballards Ln, North Finchley, London",
                "N12 8LT"
        ],
        "AvailabilityTime" : "10am - 7pm",
        "YearOfEstablishment" : "2015",
        "Capacity" : "50",
        "ProductsSold" : "Vanilla Cakes, Chocolate Muffins, Crepes, Ice cream"
```

```
}
{
        "_id" : "FARM088",
        "StoreName" : "Farm Girl",
        "HeadChef" : "Sophie McDermont",
        "Address" : [
                "59A Portobello Rd, Notting Hill, London",
                "W11 3DB"
        ],
        "AvailabilityTime" : "9am - 5:30pm",
        "YearOfEstablishment" : "2005",
        "Capacity" : "85",
        "ProductsSold" : "Bagels, Croissants, Donuts, Chocolate Cupcakes"
}
{
        "_id" : "OLDS067",
        "StoreName" : "Old Spike Peckham",
        "HeadChef" : "Emma Wilson",
        "Address" : [
                "54 Peckham Rye, Peckham, London",
                "SE15 4JR"
        ],
        "AvailabilityTime" : "9am - 5pm",
        "YearOfEstablishment" : "2019",
        "Capacity" : "150",
        "ProductsSold" : "Cheesecake, Vanilla Cupcakes, Burgers, Wings, Coffee, Alcohol"
}
{
        "_id" : "DONO689",
        "StoreName" : "Donovan's bakehouse",
        "HeadChef" : "Sophie Kim",
        "Address" : [
                "Old Spitalfields Market, Aldgate, London",
                "E1 6QS"
        ],
        "AvailabilityTime" : "9:30am - 5pm",
        "YearOfEstablishment" : "2017",
        "Capacity" : "80",
        "ProductsSold" : "Bagels, Bread, Chocolate Chip Cookies, Brownies"
}
{
        "_id" : "CAFE802",
        "StoreName" : "Cafe East",
        "HeadChef" : "Daniel Rodriguez",
        "Address" : [
                "426 Roman Rd, Bow, London",
                "E3 5LU"
        ],
        "AvailabilityTime" : "8:30am - 5pm",
        "YearOfEstablishment" : "2019",
        "Capacity" : "100",
        "ProductsSold" : "Full English Breakfast, Pancakes, Coffee, Tea, Donuts"
}
```

The following query renames one of the fields to change the spelling format and delete the duplicate misspelt field for 'ProductsSOld':

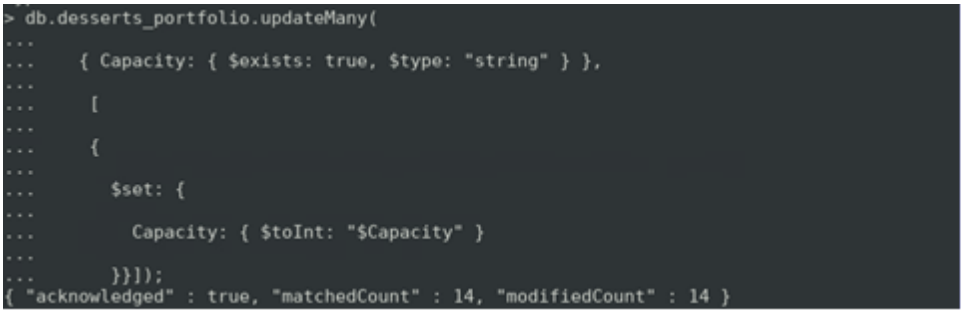
```
> db.desserts_portfolio.updateOne(
...     { _id: "GAILS007" },
...     {
...        $rename: { "YearOfEStablishment": "YearOfEstablishment" },
...        $unset: { ProductsSOld: "" }
...     }
... )
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
```

```
{
        "_id" : "GAILS007",
        "StoreName" : "GAIL's",
        "HeadChef" : "Cassie Best",
        "Address" : [
                "24 - 25 Nelson Rd, Greenwich, London",
                "SE10 9JB"
        ],
        "AvailabilityTime" : "6:30am - 8pm",
        "Capacity" : "110",
        "ProductsSold" : "Red Velvet Cake, Crossaints, Chocolate Chip Cupcakes, Brownies",
        "YearOfEstablishment" : "2006"
}
```


This query performs a text search for the word “Coffee” throughout the chosen collection. Before this query, I created a text index for the ProductSold field to enable the $text search:

```
> db.desserts_portfolio.createIndex({ ProductsSold: "text" })
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "ok" : 1
}
> db.desserts_portfolio.find({ $text: { $search: "Coffee" } })
{ "_id" : "THEH223", "StoreName" : "The Hummingbird", "HeadChef" : "Jessica Carter", "Address" : [ "47 Old Brompton Rd, South Kensington, Lon
don", "SW7 3JP" ], "AvailabilityTime" : "8am - 7pm", "YearOfEstablishment" : "2018", "Capacity" : "92", "ProductsSold" : "Cheesecake, Donuts,
 Cupcakes, Coffee" }
{ "_id" : "AOJP302", "StoreName" : "Afters Orginal", "HeadChef" : "James Paterson", "Address" : [ "166 Green St, Forest Gate, London", "E7 8J
T" ], "AvailabilityTime" : "3pm - 1am", "YearOfEstablishment" : "2009", "Capacity" : "200", "ProductsSold" : "Coffee, Red Velvet Cake, Crepes
, Waffles" }
{ "_id" : "CAFE802", "StoreName" : "Cafe East", "HeadChef" : "Daniel Rodriguez", "Address" : [ "426 Roman Rd, Bow, London", "E3 5LU" ], "Avai
labilityTime" : "8:30am - 5pm", "YearOfEstablishment" : "2019", "Capacity" : "100", "ProductsSold" : "Full English Breakfast, Pancakes, Coffe
e, Tea, Donuts" }
{ "_id" : "OLDS067", "StoreName" : "Old Spike Peckham", "HeadChef" : "Emma Wilson", "Address" : [ "54 Peckham Rye, Peckham, London", "SE15 4J
R" ], "AvailabilityTime" : "9am - 5pm", "YearOfEstablishment" : "2019", "Capacity" : "150", "ProductsSold" : "Cheesecake, Vanilla Cupcakes, B
urgers, Wings, Coffee, Alcohol" }
>
```


I noticed that Capacity was inserted as a string as opposed to an integer which is more of an appropriate field type for the value, so I converted this:



```
> db.desserts_portfolio.updateMany(
...
...    { Capacity: { $exists: true, $type: "string" } },
...
...     [
...
...     {
...
...       $set: {
...
...         Capacity: { $toInt: "$Capacity" }
...
...       }}]);
{ "acknowledged" : true, "matchedCount" : 14, "modifiedCount" : 14 }
```


The result of the previous query below. Note, the ‘Capacity’ value has been converted from a string to an integer:


```
{
        "_id" : "ZBAR574",
        "StoreName" : "Z Bar",
        "HeadChef" : "Michael Thompson",
        "Address" : [
                "58 Stoke Newington High St, Stoke Newington, London",
                "N16 7PB"
        ],
        "AvailabilityTime" : "7am - 6:30pm",
        "YearOfEstablishment" : "2013",
        "Capacity" : 100,
        "ProductsSold" : "Apple Pie, Chocolate Cakes, Crepes, Waffles"
}
```

The following query filters through the collection for stores with a capacity below 100 people:

```
> db.desserts_portfolio.find({Capacity:{$lt:100}}).pretty()
{
        "_id" : "BEIG002",
        "StoreName" : "Beigel Bake",
        "HeadChef" : "Adam Pearson",
        "Address" : [
                "159 Brick Lane, Whitechapel, London",
                "E1 6SB"
        ],
        "AvailabilityTime" : "Open 24 hours",
        "YearOfEstablishment" : "2011",
        "Capacity" : 45,
        "ProductsSold" : "Croissants, Vanilla Cupcakes, Chocolate Waffles, Filled Bagels"
}
{
        "_id" : "CAKEB019",
        "StoreName" : "Cake Box",
        "HeadChef" : "Charlotte Baker",
        "Address" : [
                "333 Ballards Ln, North Finchley, London",
                "N12 8LT"
        ],
        "AvailabilityTime" : "10am - 7pm",
        "YearOfEstablishment" : "2015",
        "Capacity" : 50,
        "ProductsSold" : "Vanilla Cakes, Chocolate Muffins, Crepes, Ice cream"
}
{
        "_id" : "DONO689",
        "StoreName" : "Donovan's bakehouse",
        "HeadChef" : "Sophie Kim",
        "Address" : [
                "Old Spitalfields Market, Aldgate, London",
                "E1 6QS"
        ],
        "AvailabilityTime" : "9:30am - 5pm",
        "YearOfEstablishment" : "2017",
        "Capacity" : 80,
        "ProductsSold" : "Bagels, Bread, Chocolate Chip Cookies, Brownies"
}
```

```
{
        "_id" : "FARM088",
        "StoreName" : "Farm Girl",
        "HeadChef" : "Sophie McDermont",
        "Address" : [
                "59A Portobello Rd, Notting Hill, London",
                "W11 3DB"
        ],
        "AvailabilityTime" : "9am - 5:30pm",
        "YearOfEstablishment" : "2005",
        "Capacity" : 85,
        "ProductsSold" : "Bagels, Croissants, Donuts, Chocolate Cupcakes"
}
{
        "_id" : "THEH223",
        "StoreName" : "The Hummingbird",
        "HeadChef" : "Jessica Carter",
        "Address" : [
                "47 Old Brompton Rd, South Kensington, London",
                "SW7 3JP"
        ],
        "AvailabilityTime" : "8am - 7pm",
        "YearOfEstablishment" : "2018",
        "Capacity" : 92,
        "ProductsSold" : "Cheesecake, Donuts, Cupcakes, Coffee"
}
```

Next, the query filters through stores that have a capacity between 150 and 300 people:

```
> db.desserts_portfolio.find({Capacity:{$gt:150,$lt:300}}).pretty()
{
        "_id" : "HEAVE422",
        "StoreName" : "Heavenly Desserts",
        "HeadChef" : "Leila Al-Farsi",
        "Address" : [
                "3 - 4 Endeavour Square, Stratford, London",
                "E15 1JN"
        ],
        "AvailabilityTime" : "12pm - 11pm",
        "YearOfEstablishment" : "2020",
        "Capacity" : 180,
        "ProductsSold" : "Baklava, Donuts, Mazurek, Cinnamon Rolls"
}
{
        "_id" : "AOJP302",
        "StoreName" : "Afters Orginal",
        "HeadChef" : "James Paterson",
        "Address" : [
                "166 Green St, Forest Gate, London",
                "E7 8JT"
        ],
        "AvailabilityTime" : "3pm - 1am",
        "YearOfEstablishment" : "2009",
        "Capacity" : 200,
        "ProductsSold" : "Coffee, Red Velvet Cake, Crepes, Waffles"
}
```

I updated the YearOfEstablishment which was listed as a string, to an integer as this is better tailored:

```
> db.desserts_portfolio.updateMany(
... {YearOfEstablishment: {$exists: true, $type: "string"}},
...       [
...            {
...               $set:  {
...                  YearOfEstablishment: {$toInt: "$YearOfEstablishment"
...              }}}])
{ "acknowledged" : true, "matchedCount" : 14, "modifiedCount" : 14 }
```

This query searches for stores that opened after the year 2010:

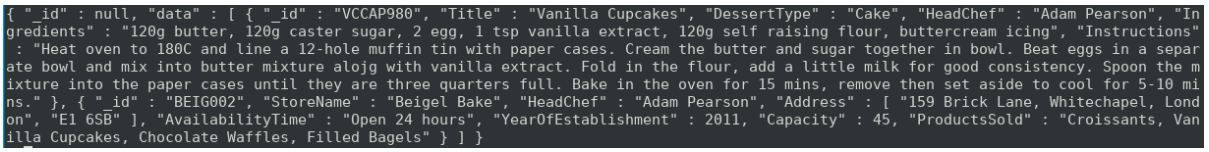
```
> db.desserts_portfolio.find({YearOfEstablishment:{$gt:2010}}).pretty()
{
        "_id" : "HEAVE422",
        "StoreName" : "Heavenly Desserts",
        "HeadChef" : "Leila Al-Farsi",
        "Address" : [
                "3 - 4 Endeavour Square, Stratford, London",
                "E15 1JN"
        ],
        "AvailabilityTime" : "12pm - 11pm",
        "YearOfEstablishment" : 2020,
        "Capacity" : 180,
        "ProductsSold" : "Baklava, Donuts, Mazurek, Cinnamon Rolls"
}
{
        "_id" : "ZBAR574",
        "StoreName" : "Z Bar",
        "HeadChef" : "Michael Thompson",
        "Address" : [
                "58 Stoke Newington High St, Stoke Newington, London",
                "N16 7PB"
        ],
        "AvailabilityTime" : "7am - 6:30pm",
        "YearOfEstablishment" : 2013,
        "Capacity" : 100,
        "ProductsSold" : "Apple Pie, Chocolate Cakes, Crepes, Waffles"
}
```

```
{
        "_id" : "BEIG002",
        "StoreName" : "Beigel Bake",
        "HeadChef" : "Adam Pearson",
        "Address" : [
                "159 Brick Lane, Whitechapel, London",
                "E1 6SB"
        ],
        "AvailabilityTime" : "Open 24 hours",
        "YearOfEstablishment" : 2011,
        "Capacity" : 45,
        "ProductsSold" : "Croissants, Vanilla Cupcakes, Chocolate Waffles, Filled Bagels"
}
{
        "_id" : "THEH223",
        "StoreName" : "The Hummingbird",
        "HeadChef" : "Jessica Carter",
        "Address" : [
                "47 Old Brompton Rd, South Kensington, London",
                "SW7 3JP"
        ],
        "AvailabilityTime" : "8am - 7pm",
        "YearOfEstablishment" : 2018,
        "Capacity" : 92,
        "ProductsSold" : "Cheesecake, Donuts, Cupcakes, Coffee"
}
{
        "_id" : "CAKEB019",
        "StoreName" : "Cake Box",
        "HeadChef" : "Charlotte Baker",
        "Address" : [
                "333 Ballards Ln, North Finchley, London",
                "N12 8LT"
        ],
        "AvailabilityTime" : "10am - 7pm",
        "YearOfEstablishment" : 2015,
        "Capacity" : 50,
        "ProductsSold" : "Vanilla Cakes, Chocolate Muffins, Crepes, Ice cream"
}
```



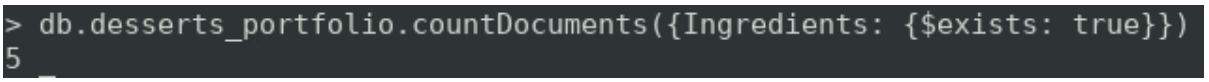
```
{
        "_id" : "OLDS067",
        "StoreName" : "Old Spike Peckham",
        "HeadChef" : "Emma Wilson",
        "Address" : [
                "54 Peckham Rye, Peckham, London",
                "SE15 4JR"
        ],
        "AvailabilityTime" : "9am - 5pm",
        "YearOfEstablishment" : 2019,
        "Capacity" : 150,
        "ProductsSold" : "Cheesecake, Vanilla Cupcakes, Burgers, Wings, Coffee, Alcohol"
}
{
        "_id" : "DONO689",
        "StoreName" : "Donovan’s bakehouse",
        "HeadChef" : "Sophie Kim",
        "Address" : [
                "Old Spitalfields Market, Aldgate, London",
                "E1 6QS"
        ],
        "AvailabilityTime" : "9:30am - 5pm",
        "YearOfEstablishment" : 2017,
        "Capacity" : 80,
        "ProductsSold" : "Bagels, Bread, Chocolate Chip Cookies, Brownies"
}
{
        "_id" : "CAFE802",
        "StoreName" : "Cafe East",
        "HeadChef" : "Daniel Rodriguez",
        "Address" : [
                "426 Roman Rd, Bow, London",
                "E3 5LU"
        ],
        "AvailabilityTime" : "8:30am - 5pm",
        "YearOfEstablishment" : 2019,
        "Capacity" : 100,
        "ProductsSold" : "Full English Breakfast, Pancakes, Coffee, Tea, Donuts"
}
```

This following query will self-join the collection to group documents associated with the Head Chef "Adam Pearson." This can be useful for users looking for information regarding desserts and stores related to Adam:

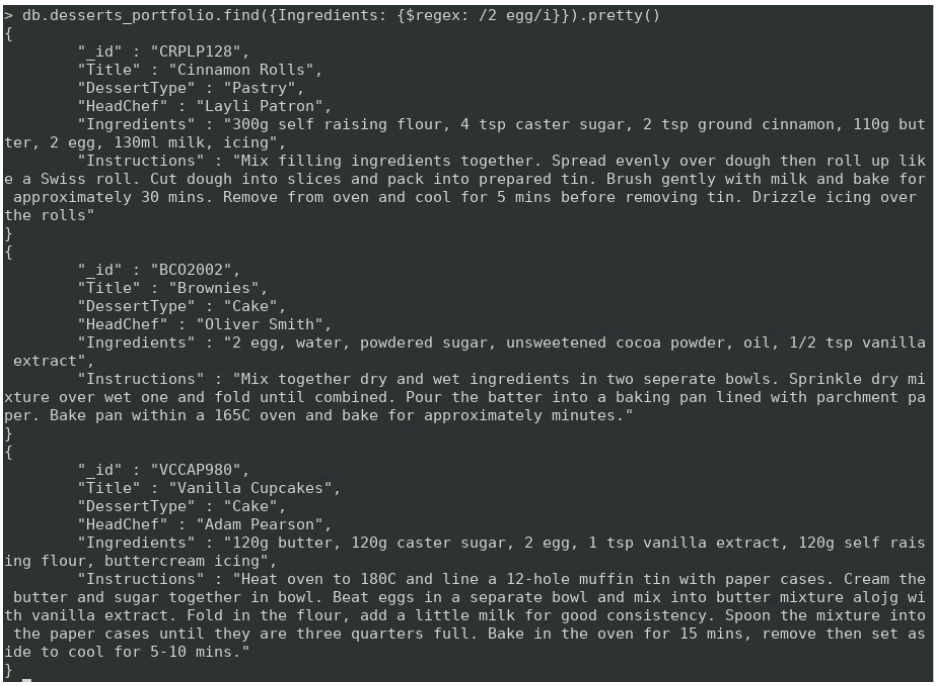


The following query searches for all stores with a postcode starting with E1, in case users are looking for stores within specified regions, accounting for case sensitivity:

This query counts the number of recipes that exist within the document by searching the field 'Ingredients' which is unique to dessert products:



This query searches for documents that include the field 'Ingredients', specifically looking for the string value '2 eggs:'



The query below runs an aggregation filter to single out documents that have a Capacity over 90. The next step singles out the specified fields and creates a new field ‘Revenue’ which provides the output of capacity multiplied by 5 - useful to calculate estimated revenue:

```
> db.desserts_portfolio.aggregate([
...   {
...     $match: {
...       Capacity: {$gt: 90}}
...   },
... {
... $project: {
...    _id: 1,
...    StoreName: 1,
...    Capacity: 1,
...    Revenue: {$multiply: ["$Capacity", 5]}}
... }
... ])
{ "_id" : "THEH223", "StoreName" : "The Hummingbird", "Capacity" : 92, "Revenue" : 460 }
{ "_id" : "ZBAR574", "StoreName" : "Z Bar", "Capacity" : 100, "Revenue" : 500 }
{ "_id" : "CAFE802", "StoreName" : "Cafe East", "Capacity" : 100, "Revenue" : 500 }
{ "_id" : "GAILS007", "StoreName" : "GAIL's", "Capacity" : 110, "Revenue" : 550 }
{ "_id" : "PARKS800", "StoreName" : "Parkside Cafe", "Capacity" : 110, "Revenue" : 550 }
{ "_id" : "THEB001", "StoreName" : "The Bakery", "Capacity" : 120, "Revenue" : 600 }
{ "_id" : "OLDS067", "StoreName" : "Old Spike Peckham", "Capacity" : 150, "Revenue" : 750 }
{ "_id" : "HEAVE422", "StoreName" : "Heavenly Desserts", "Capacity" : 180, "Revenue" : 900 }
{ "_id" : "AOJP302", "StoreName" : "Afters Orginal", "Capacity" : 200, "Revenue" : 1000 }
{ "_id" : "CASH418", "StoreName" : "Cakes & Shakes", "Capacity" : 300, "Revenue" : 1500 }
```