Computer Engineering Department

Course Name: CMPE 272

Team No: 3

Student Names: Ambika Bohra, Lavanya Kandukuri, Naresh Kumar,Sai Anusha Mulumoodi

Instructor: Rakesh Ranjan

Assignment #1: Database Team based Assignment

SQLite : Purchase Order Management System

Schema:

Customer : cust_id (PK),cust_name, cust_address, cust_phone
Order : order_id(PK) , item_id, item_quantity, order_date, total_bill, cust_id(FK)
Item : item_id (PK), item _name, item_price, category_id(FK)
Category : category_id(PK), category_desc

Create Statement:

Created above tables using option " add table" in SQLite

CREATE TABLE "main"."customer" ("cust_id" INTEGER PRIMARY KEY NOT NULL , "cust_name" VARCHAR NOT NULL , "cust_address" VARCHAR, "cust_phone" INTEGER)

CREATE TABLE "main"."category" ("category_id" INTEGER PRIMARY KEY NOT NULL , "category_desc" VARCHAR NOT NULL)

CREATE TABLE "main"."item" ("item_id" INTEGER PRIMARY KEY NOT NULL , "item_name" VARCHAR, "item_price" DOUBLE NOT NULL DEFAULT 0, "category_id" INTEGER NOT NULL, FOREIGN KEY(category_id) REFERENCES category(category_id))

CREATE TABLE "main"."order" ("order_id" INTEGER PRIMARY KEY NOT NULL , "item_id" INTEGER NOT NULL , "item_qty" INTEGER NOT NULL DEFAULT 1, "total_bill" DOUBLE NOT NULL DEFAULT 0, "order_date" DATETIME NOT NULL DEFAULT CURRENT_DATE, "cust_id" INTEGER NOT NULL , FOREIGN KEY(cust_id) REFERENCES customer(cust_id), FOREIGN KEY(item_id) REFERENCES item(item_id))

Insert Query:

Inserted sample data into these tables using :add row" option in SQLite

INSERT INTO "main"."**customer**" ("cust_id","cust_name","cust_address","cust_phone") VALUES (100,?1,?2,?3)
Parameters:

param 1 (text): john
param 2 (text): e-23, Link road, CA
param 3 (integer): 9873212345

INSERT INTO "main"."**category**" ("category_id","category_desc") VALUES (2,?1)
Parameters:
param 1 (text): electronics

INSERT INTO "main"."**item**" ("item_name","item_price","category_id") VALUES (?1,?2,?3)
Parameters:
param 1 (text): mattress
param 2 (integer): 400
param 3 (integer): 1

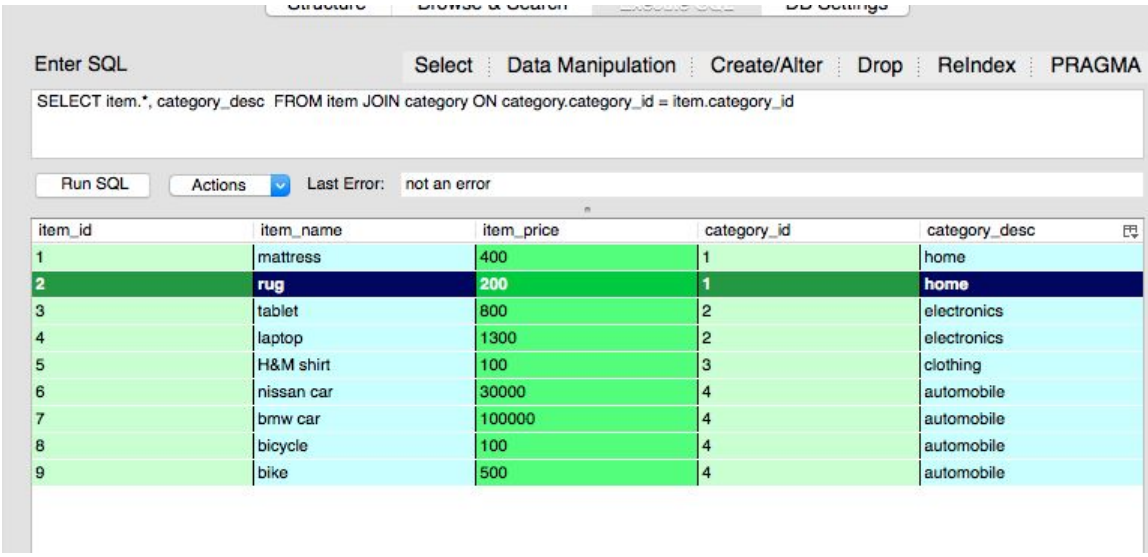
INSERT INTO "main"."**order**" ("order_id","item_id","order_date","cust_id") VALUES (1001,?1,?2,?3)
Parameters:
param 1 (integer): 1
param 2 (text): 2017/02/18
param 3 (integer): 100

Following is the sample error statement thrown by SQLite in case of violation of primary key.

SQLiteManager: Execute failed: INSERT INTO "main"."order" ("order_id","item_id","order_date","cust_id") VALUES (1001,?1,?2,?3) [UNIQUE constraint failed: order.order_id]
Exception Name: NS_ERROR_STORAGE_CONSTRAINT
Exception Message: Component returned failure code: 0x80630003 (NS_ERROR_STORAGE_CONSTRAINT) [mozIStorageStatement.execute]

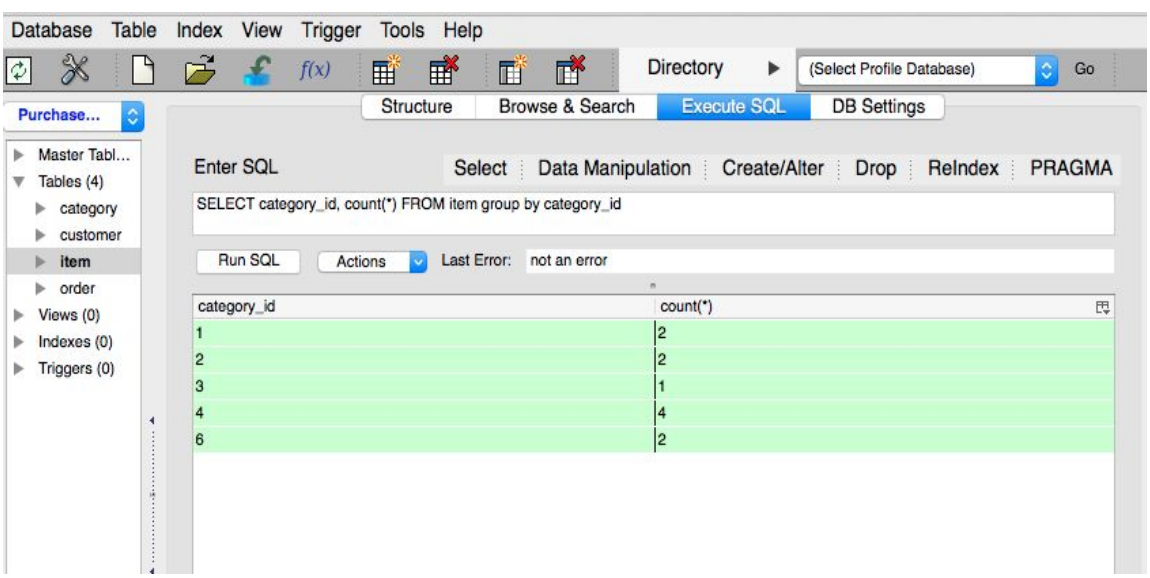
Queries:

1) Join Query : (category and item)



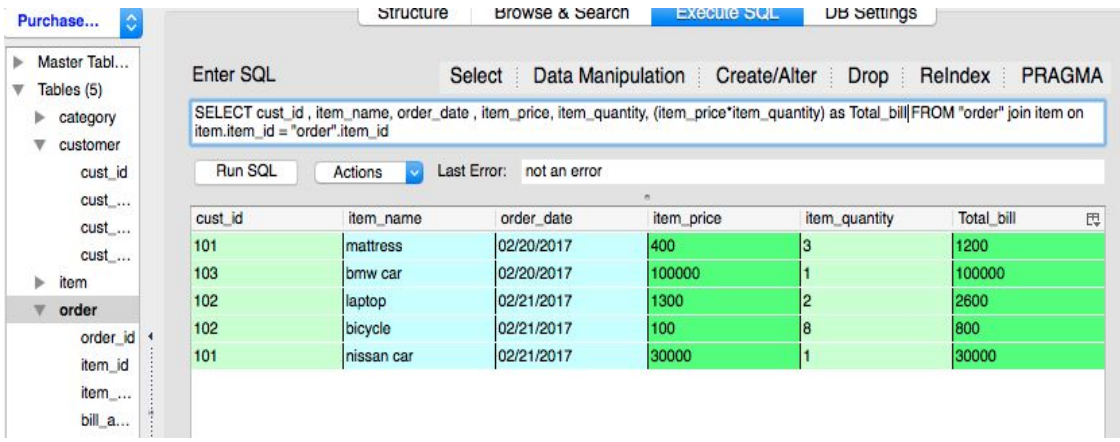
item_id	item_name	item_price	category_id	category_desc
1	mattress	400	1	home
2	rug	200	1	home
3	tablet	800	2	electronics
4	laptop	1300	2	electronics
5	H&M shirt	100	3	clothing
6	nissan car	30000	4	automobile
7	bmw car	100000	4	automobile
8	bicycle	100	4	automobile
9	bike	500	4	automobile

2) “GROUP BY“ for counting items count of different categories :



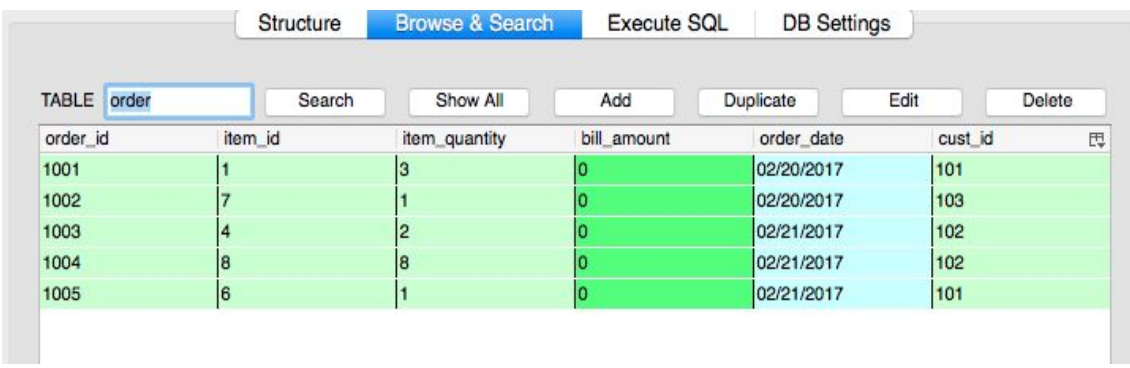
category_id	count(*)
1	2
2	2
3	1
4	4
6	2

3) Join Query(item and order) to get bill amount :



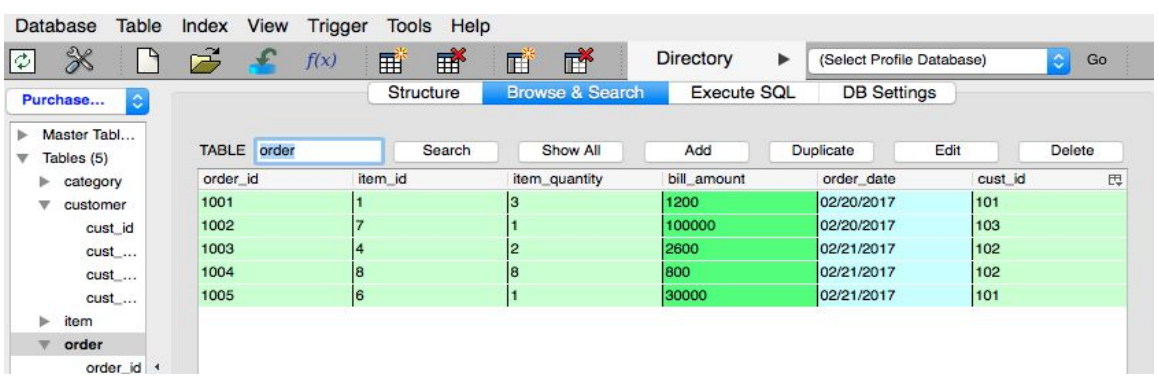
cust_id	item_name	order_date	item_price	item_quantity	Total_bill
101	mattress	02/20/2017	400	3	1200
103	bmw car	02/20/2017	100000	1	100000
102	laptop	02/21/2017	1300	2	2600
102	bicycle	02/21/2017	100	8	800
101	nissan car	02/21/2017	30000	1	30000

4) Update Statement to update bill amount
(according to item_price from item table and item quantity selected by customer)



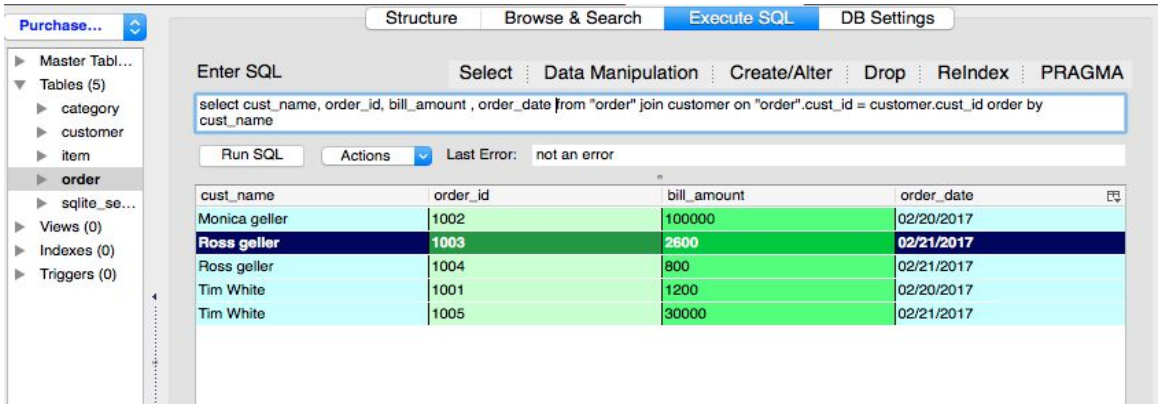
order_id	item_id	item_quantity	bill_amount	order_date	cust_id
1001	1	3	0	02/20/2017	101
1002	7	1	0	02/20/2017	103
1003	4	2	0	02/21/2017	102
1004	8	8	0	02/21/2017	102
1005	6	1	0	02/21/2017	101

update "order" set bill_amount = item_quantity * (select item_price from item where item.item_id = "order".item_id)



order_id	item_id	item_quantity	bill_amount	order_date	cust_id
1001	1	3	1200	02/20/2017	101
1002	7	1	100000	02/20/2017	103
1003	4	2	2600	02/21/2017	102
1004	8	8	800	02/21/2017	102
1005	6	1	30000	02/21/2017	101

5) order by query :



DB2 Express C:

1. After downloading DB2 express C, a sample database named “SAMPLE” is created by using the command “**db2sampl**”.

Output:

```
[db2inst1@4ddade5639fb ~]$ db2sampl

  Creating database "SAMPLE"...
  Connecting to database "SAMPLE"...
  Creating tables and data in schema "DB2INST1"...
  Creating tables with XML columns and XML data in schema "DB2INST1"...

  'db2sampl' processing complete.
```

2. A sample query using where clause and Group by is run for which the output looks like:

Sample Query:

db2 select WORKDEPT,SALARY from EMPLOYEE **where** BONUS = 800.00 **group by** WORKDEPT,SALARY

Output:

```
[db2inst1@9524da63b3cb ~]$ db2 select WORKDEPT,SALARY from EMPLOYEE where BONUS = 800.00 group by WORKDEPT,SALARY;

WORKDEPT SALARY
-------- -----------
E01         80175.00
B01         94250.00
C01         98250.00

  3 record(s) selected.
```

3. Query explain plan using **db2exfmt** tool is generated as follows:

Query that is used is:

select EMPNO,WORKDEPT,DEPTNAME from EMPLOYEE e,DEPARTMENT d where e.WORKDEPT=d.DEPTNO;

Query Snapshot:

```
[db2inst1@9524da63b3cb misc]$ db2 explain plan with snapshot for select EMPNO,WORKDEPT,DEPTNAME from EMPLOYEE e,DEPARTMENT d where e.WORKDEPT=d.DEPTNO;
DB20000I  The SQL command completed successfully.
```

Db2exfmt Output Snapshot:

```
[db2inst1@9524da63b3cb misc]$ db2exfmt
DB2 Universal Database Version 10.5, 5622-044 (c) Copyright IBM Corp. 1991, 2012
Licensed Material - Program Property of IBM
IBM DATABASE 2 Explain Table Format Tool

Enter Database Name ==> SAMPLE
Connecting to the Database.
Connect to Database Successful.
Binding package - Bind was Successful
Enter up to 26 character Explain timestamp (Default -1) ==>
Enter up to 128 character source name (SOURCE_NAME, Default %%) ==>
Enter source schema (SOURCE_SCHEMA, Default %%) ==>
Enter section number (0 for all, Default 0) ==>
Enter outfile name. Default is to terminal ==>
DB2 Universal Database Version 10.5, 5622-044 (c) Copyright IBM Corp. 1991, 2012
Licensed Material - Program Property of IBM
IBM DATABASE 2 Explain Table Format Tool
```

```
******************** EXPLAIN INSTANCE ********************

DB2_VERSION:       10.05.5
FORMATTED ON DB:   SAMPLE
SOURCE_NAME:       SQLC2K26
SOURCE_SCHEMA:     NULLID
SOURCE_VERSION:
EXPLAIN_TIME:      2017-02-28-07.29.04.350134
EXPLAIN_REQUESTER: DB2INST1

Database Context:
----------------
        Parallelism:          None
        CPU Speed:            1.889377e-07
        Comm Speed:           0
        Buffer Pool size:     1000
        Sort Heap size:       256
        Database Heap size:   1200
        Lock List size:       4096
        Maximum Lock List:    10
        Average Applications: 1
        Locks Available:      13107

Package Context:
---------------
        SQL Type:           Dynamic
        Optimization Level: 5
        Blocking:           Block All Cursors
        Isolation Level:    Cursor Stability


---------------- STATEMENT 1  SECTION 203 ----------------
        QUERYNO:       1
        QUERYTAG:
        Statement Type:   Select
        Updatable:        No
        Deletable:        No
        Query Degree:     1

Original Statement:
------------------
select
  EMPNO,
  WORKDEPT,
  DEPTNAME
from
  EMPLOYEE e,
  DEPARTMENT d
where
  e.WORKDEPT=d.DEPTNO
```

```
Optimized Statement:
--------------------
SELECT
  Q2.EMPNO AS "EMPNO",
  Q2.WORKDEPT AS "WORKDEPT",
  Q1.DEPTNAME AS "DEPTNAME"
FROM
  DB2INST1.DEPARTMENT AS Q1,
  DB2INST1.EMPLOYEE AS Q2
WHERE
  (Q2.WORKDEPT = Q1.DEPTNO)

Access Plan:
------------
        Total Cost:             13.6503
        Query Degree:           1

              Rows
             RETURN
             (   1)
              Cost
               I/O
               |
               42
             ^HSJOIN
             (   2)
             13.6503
                2
         /-----+------\
       42               14
     TBSCAN           TBSCAN
     (   3)           (   4)
     6.82855          6.81944
        1                1
       |                |
       42               14
 TABLE: DB2INST1  TABLE: DB2INST1
    EMPLOYEE        DEPARTMENT
       Q2               Q1


Operator Symbols :
------------------

   Symbol      Description
   ---------   -----------------------------------------
   >JOIN     : Left outer join
    JOIN<    : Right outer join
   >JOIN<    : Full outer join
   xJOIN     : Left antijoin
    JOINx    : Right antijoin
   ^JOIN     : Left early out
    JOIN^    : Right early out
```

```
Extended Diagnostic Information:
--------------------------------

Diagnostic Identifier:  1
Diagnostic Details:     EXP0062W  The following MQT or statistical view was
                        not eligible because one or more columns or
                        expressions referenced in the query were not found
                        in the MQT: "DB2INST1"."ADEFUSR".
Diagnostic Identifier:  2
Diagnostic Details:     EXP0148W  The following MQT or statistical view was
                        considered in query matching: "DB2INST1"."ADEFUSR".

Plan Details:
-------------


        1) RETURN: (Return Result)
                Cumulative Total Cost:          13.6503
                Cumulative CPU Cost:            213256
                Cumulative I/O Cost:            2
                Cumulative Re-Total Cost:       13.6503
                Cumulative Re-CPU Cost:         213256
                Cumulative Re-I/O Cost:         2
                Cumulative First Row Cost:      13.6503
                Estimated Bufferpool Buffers:   1

                Arguments:
                ---------
                BLDLEVEL: (Build level)
                        DB2 v10.5.0.5 : s141128
                HEAPUSE : (Maximum Statement Heap Usage)
                        96 Pages
                PLANID  : (Access plan identifier)
                        aefec680bca855f3
                PREPTIME: (Statement prepare time)
                                150 milliseconds
                SEMEVID : (Semantic environment identifier)
                        431f78d03d9bb07e
                STMTHEAP: (Statement heap size)
                        8192
                STMTID  : (Normalized statement identifier)
                        39d4f22e846d81ea

                Input Streams:
                -------------
                        5) From Operator #2

                                Estimated number of rows:       42
                                Number of columns:              3
                                Subquery predicate ID:          Not Applicable

                                Column Names:
                                ------------
                                +Q3.DEPTNAME+Q3.WORKDEPT+Q3.EMPNO
```

```
2) HSJOIN: (Hash Join)
        Cumulative Total Cost:          13.6503
        Cumulative CPU Cost:            213256
        Cumulative I/O Cost:            2
        Cumulative Re-Total Cost:       13.6503
        Cumulative Re-CPU Cost:         213256
        Cumulative Re-I/O Cost:         2
        Cumulative First Row Cost:      13.6503
        Estimated Bufferpool Buffers:   1

        Arguments:
        ---------
        BITFLTR : (Hash Join Bit Filter used)
                FALSE
        EARLYOUT: (Early Out flag)
                LEFT
        HASHCODE: (Hash Code Size)
                24 BIT
        HASHTBSZ: (Number of hash table entries)
                14
        TEMPSIZE: (Temporary Table Page Size)
                8192
        TUPBLKSZ: (Tuple Block Size (bytes))
                4000

        Predicates:
        ----------
        2) Predicate used in Join,
                Comparison Operator:            Equal (=)
                Subquery Input Required:        No
                Filter Factor:                  0.0714286

                Predicate Text:
                --------------
                (Q2.WORKDEPT = Q1.DEPTNO)


        Input Streams:
        -------------
                2) From Operator #3

                        Estimated number of rows:       42
                        Number of columns:              2
                        Subquery predicate ID:          Not Applicable

                        Column Names:
                        ------------
                        +Q2.EMPNO+Q2.WORKDEPT

                4) From Operator #4

                        Estimated number of rows:       14
```

```
                Number of columns:              2
                Subquery predicate ID:          Not Applicable

                Column Names:
                ------------
                +Q1.DEPTNAME+Q1.DEPTNO


        Output Streams:
        --------------
                5) To Operator #1

                Estimated number of rows:       42
                Number of columns:              3
                Subquery predicate ID:          Not Applicable

                Column Names:
                ------------
                +Q3.DEPTNAME+Q3.WORKDEPT+Q3.EMPNO


3) TBSCAN: (Table Scan)
        Cumulative Total Cost:          6.82855
        Cumulative CPU Cost:            124639
        Cumulative I/O Cost:            1
        Cumulative Re-Total Cost:       0.0139901
        Cumulative Re-CPU Cost:         74046
        Cumulative Re-I/O Cost:         0
        Cumulative First Row Cost:      6.81488
        Estimated Bufferpool Buffers:   1

        Arguments:
        ---------
        CUR_COMM: (Currently Committed)
                TRUE
        JN INPUT: (Join input leg)
                OUTER
        LCKAVOID: (Lock Avoidance)
                TRUE
        MAXPAGES: (Maximum pages for prefetch)
                ALL
        PREFETCH: (Type of Prefetch)
                NONE
        ROWLOCK : (Row Lock intent)
                SHARE (CS/RS)
        SCANDIR : (Scan Direction)
                FORWARD
        SKIP_INS: (Skip Inserted Rows)
                TRUE
        SPEED   : (Assumed speed of scan, in sharing structures)
                FAST
        TABLOCK : (Table Lock intent)
                INTENT SHARE
        TBISOLVL: (Table access Isolation Level)
```

```
                CURSOR STABILITY
        THROTTLE: (Scan may be throttled, for scan sharing)
                TRUE
        VISIBLE : (May be included in scan sharing structures)
                TRUE
        WRAPPING: (Scan may start anywhere and wrap)
                TRUE

        Input Streams:
        -------------
                1) From Object DB2INST1.EMPLOYEE

                        Estimated number of rows:       42
                        Number of columns:              3
                        Subquery predicate ID:          Not Applicable

                        Column Names:
                        ------------
                        +Q2.$RID$+Q2.EMPNO+Q2.WORKDEPT


        Output Streams:
        --------------
                2) To Operator #2

                        Estimated number of rows:       42
                        Number of columns:              2
                        Subquery predicate ID:          Not Applicable

                        Column Names:
                        ------------
                        +Q2.EMPNO+Q2.WORKDEPT


4) TBSCAN: (Table Scan)
        Cumulative Total Cost:          6.81944
        Cumulative CPU Cost:            76423
        Cumulative I/O Cost:            1
        Cumulative Re-Total Cost:       0.00488026
        Cumulative Re-CPU Cost:         25830
        Cumulative Re-I/O Cost:         0
        Cumulative First Row Cost:      6.81488
        Estimated Bufferpool Buffers:   1

        Arguments:
        ---------
        CUR_COMM: (Currently Committed)
                TRUE
        JN INPUT: (Join input leg)
                INNER
        LCKAVOID: (Lock Avoidance)
                TRUE
        MAXPAGES: (Maximum pages for prefetch)
```

```
                ALL
        PREFETCH: (Type of Prefetch)
                NONE
        ROWLOCK : (Row Lock intent)
                SHARE (CS/RS)
        SCANDIR : (Scan Direction)
                FORWARD
        SKIP_INS: (Skip Inserted Rows)
                TRUE
        SPEED   : (Assumed speed of scan, in sharing structures)
                FAST
        TABLOCK : (Table Lock intent)
                INTENT SHARE
        TBISOLVL: (Table access Isolation Level)
                CURSOR STABILITY
        THROTTLE: (Scan may be throttled, for scan sharing)
                TRUE
        VISIBLE : (May be included in scan sharing structures)
                TRUE
        WRAPPING: (Scan may start anywhere and wrap)
                TRUE

        Input Streams:
        -------------
                3) From Object DB2INST1.DEPARTMENT

                        Estimated number of rows:       14
                        Number of columns:              3
                        Subquery predicate ID:          Not Applicable

                        Column Names:
                        ------------
                        +Q1.$RID$+Q1.DEPTNAME+Q1.DEPTNO


        Output Streams:
        --------------
                4) To Operator #2

                        Estimated number of rows:       14
                        Number of columns:              2
                        Subquery predicate ID:          Not Applicable

                        Column Names:
                        ------------
                        +Q1.DEPTNAME+Q1.DEPTNO


Objects Used in Access Plan:
----------------------------

        Schema: DB2INST1
        Name:   ADEFUSR
        Type:   Materialized View (reference only)


        Schema: DB2INST1
        Name:   DEPARTMENT
        Type:   Table

        Schema: DB2INST1
        Name:   EMPLOYEE
        Type:   Table


Extended Statistics Information:
--------------------------------

Tablespace Context:
-------------------
                Name:                                   USERSPACE1
                Overhead:                               6.725000
                Transfer Rate:                          0.080000
                Prefetch Size:                          32
                Extent Size:                            32
                Type:                                   Database managed
                Partition Group Name:                   NULLP
                Buffer Pool Identifier:                 0

Base Table Statistics:
----------------------
Name  :         EMPLOYEE
Schema:         DB2INST1
                Number of Columns:                      14
                Number of Pages with Rows:              1
                Number of Meta Pages:                   0
                Number of Pages:                        1
                Number of Rows:                         42
                Table Overflow Record Count:            0
                Width of Rows:                          40
                Time of Creation:                       2017-02-28-06.26.07.076509
                Last Statistics Update:                 2017-02-28-06.35.36.360000
                Primary Tablespace:                     USERSPACE1
                Tablespace for Indexes:                 USERSPACE1
                Tablespace for Long Data:               NULLP
                Number of Referenced Columns:           2
                Number of Indexes:                      2
                Volatile Table:                         No
                Number of Active Blocks:                -1
                Number of Column Groups:                0
                Number of Data Partitions:              1
                Average Row Compression Ratio:          -1.000000
                Percent Rows Compressed:                -1.000000
                Average Compressed Row Size:            -1
                Statistics Type:                        A

        Column Information:
        -------------------
                Number:                                 1
```

```
        Name:                                           EMPNO
        Statistics Available:                           Yes

        Column Statistics:
        ------------------
                Schema name of the column type:                 SYSIBM
                Name of column type:                            CHARACTER
                Maximum column length:                          6
                Scale for decimal or timestamp column:          0
                Number of distinct column values:               42
                Average column length:                          6
                Number of most frequent values:                 -1
                Number of quantiles:                            20
                Second highest data value:                      200330
                Second lowest data value:                       000020
                Column sequence in partition key:               0
                Average number of sub-elements:                 -1
                Average length of delimiters:                   -1
                Percentage encoded column values:               -1
                Column clustering:                              -1

        Column Distribution Statistics:
        -------------------------------
                Quantile Statistics:
                Valcount        Value           Distcount
                ------------------------------------------------
                1               000010          1
                2               000020          2
                4               000050          4
                7               000090          7
                9               000110          9
                11              000130          11
                13              000150          13
                15              000170          15
                18              000200          18
                20              000220          20
                22              000240          22
                24              000260          24
                27              000290          27
                29              000310          29
                31              000330          31
                33              200010          33
                35              200140          35
                38              200240          38
                40              200310          40
                42              200340          42

Column Information:
-------------------
        Number:                                         5
        Name:                                           WORKDEPT
        Statistics Available:                           Yes

        Column Statistics:
```

```
                Schema name of the column type:                 SYSIBM
                Name of column type:                            CHARACTER
                Maximum column length:                          3
                Scale for decimal or timestamp column:          0
                Number of distinct column values:               8
                Average column length:                          4
                Number of most frequent values:                 6
                Number of quantiles:                            13
                Second highest data value:                      E11
                Second lowest data value:                       B01
                Column sequence in partition key:               0
                Average number of sub-elements:                 -1
                Average length of delimiters:                   -1
                Percentage encoded column values:               -1
                Column clustering:                              -1

        Column Distribution Statistics:
        -------------------------------
                Frequency Statistics:
                Valcount        Value
                ----------------------------------
                11              D11
                7               D21
                7               E11
                6               E21
                5               A00
                4               C01

                Quantile Statistics:
                Valcount        Value           Distcount
                -------------------------------------------------
                0               A00             1
                5               A00             1
                6               C01             3
                10              C01             3
                10              D11             4
                21              D11             4
                21              D21             5
                28              D21             5
                29              E01             6
                29              E11             7
                36              E11             7
                36              E21             8
                42              E21             8

Indexes defined on the table:
-----------------------------
Name  : XEMP2
Schema: DB2INST1
        Unique Rule:                                    Duplicate index
        Used in Operator:                               No
        Page Fetch Pairs:                               Not Available
        Number of Columns:                              1
        Index Leaf Pages:                               1
```

```
                Index Tree Levels:                              1
                Index First Key Cardinality:                    8
                Index Full Key Cardinality:                     8
                Index Cluster Ratio:                            100
                Index Cluster Factor:                           -1.000000
                Time of Creation:                               2017-02-28-06.26.08.475649
                Last Statistics Update:                         2017-02-28-06.35.36.360000
                Index Sequential Pages:                         0
                Index First 2 Keys Cardinality:                 -1
                Index First 3 Keys Cardinality:                 -1
                Index First 4 Keys Cardinality:                 -1
                Index Avg Gap between Sequences:                0.000000
                Fetch Avg Gap between Sequences:                -1.000000
                Index Avg Sequential Pages:                     0.000000
                Fetch Avg Sequential Pages:                     -1.000000
                Index Avg Random Pages:                         1.000000
                Fetch Avg Random Pages:                         -1.000000
                Index RID Count:                                42
                Index Deleted RID Count:                        0
                Index Empty Leaf Pages:                         0
                Avg Partition Cluster Ratio:                    -1
                Avg Partition Cluster Factor:                   -1.000000
                Data Partition Cluster Factor:                  1.000000
                Data Partition Page Fetch Pairs:                Not Available

        Name  : PK_EMPLOYEE
        Schema: DB2INST1
                Unique Rule:                                    Primary key index
                Used in Operator:                               No
                Page Fetch Pairs:                               Not Available
                Number of Columns:                              1
                Index Leaf Pages:                               1
                Index Tree Levels:                              1
                Index First Key Cardinality:                    42
                Index Full Key Cardinality:                     42
                Index Cluster Ratio:                            100
                Index Cluster Factor:                           -1.000000
                Time of Creation:                               2017-02-28-06.26.07.076509
                Last Statistics Update:                         2017-02-28-06.35.36.360000
                Index Sequential Pages:                         0
                Index First 2 Keys Cardinality:                 -1
                Index First 3 Keys Cardinality:                 -1
                Index First 4 Keys Cardinality:                 -1
                Index Avg Gap between Sequences:                0.000000
                Fetch Avg Gap between Sequences:                -1.000000
                Index Avg Sequential Pages:                     0.000000
                Fetch Avg Sequential Pages:                     -1.000000
                Index Avg Random Pages:                         1.000000
                Fetch Avg Random Pages:                         -1.000000
                Index RID Count:                                42
                Index Deleted RID Count:                        0
                Index Empty Leaf Pages:                         0
                Avg Partition Cluster Ratio:                    -1
                Avg Partition Cluster Factor:                   -1.000000

                Data Partition Cluster Factor:                  1.000000
                Data Partition Page Fetch Pairs:                Not Available

Base Table Statistics:
----------------------
Name  :         DEPARTMENT
Schema:         DB2INST1
                Number of Columns:                              5
                Number of Pages with Rows:                      1
                Number of Meta Pages:                           0
                Number of Pages:                                1
                Number of Rows:                                 14
                Table Overflow Record Count:                    0
                Width of Rows:                                  48
                Time of Creation:                               2017-02-28-06.26.05.534125
                Last Statistics Update:                         2017-02-28-06.40.36.588812
                Primary Tablespace:                             USERSPACE1
                Tablespace for Indexes:                         USERSPACE1
                Tablespace for Long Data:                       NULLP
                Number of Referenced Columns:                   2
                Number of Indexes:                              3
                Volatile Table:                                 No
                Number of Active Blocks:                        -1
                Number of Column Groups:                        0
                Number of Data Partitions:                      1
                Average Row Compression Ratio:                  -1.000000
                Percent Rows Compressed:                        -1.000000
                Average Compressed Row Size:                    -1
                Statistics Type:                                A

        Column Information:
        -------------------
                Number:                                         2
                Name:                                           DEPTNAME
                Statistics Available:                           Yes

                Column Statistics:
                ------------------
                        Schema name of the column type:         SYSIBM
                        Name of column type:                    VARCHAR
                        Maximum column length:                  36
                        Scale for decimal or timestamp column:  0
                        Number of distinct column values:       14
                        Average column length:                  21
                        Number of most frequent values:         -1
                        Number of quantiles:                    14
                        Second highest data value:              SPIFFY COMPUTER SERVICE DIV.
                        Second lowest data value:               BRANCH OFFICE F2
                        Column sequence in partition key:       0
                        Average number of sub-elements:         -1
                        Average length of delimiters:           -1
                        Percentage encoded column values:       -1
                        Column clustering:                      -1
```

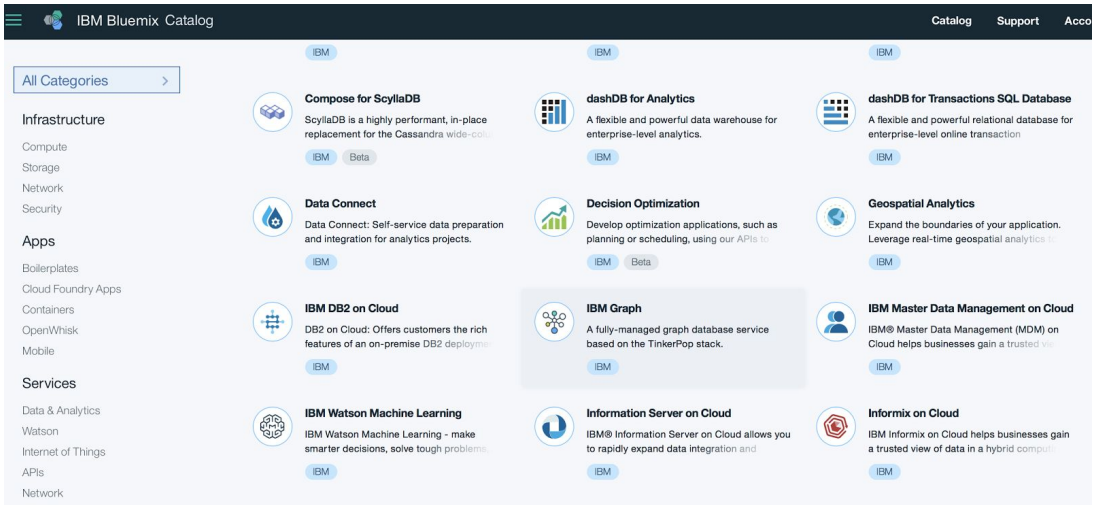
```
        Column Distribution Statistics:
        -------------------------------
                Quantile Statistics:
                Valcount        Value           Distcount
                ------------------------------------------------
                1               ADMINISTRATION SYSTEMS          0
                2               BRANCH OFFICE F2                0
                3               BRANCH OFFICE G2                0
                4               BRANCH OFFICE H2                0
                5               BRANCH OFFICE I2                0
                6               BRANCH OFFICE J2                0
                7               DEVELOPMENT CENTER              0
                8               INFORMATION CENTER              0
                10              OPERATIONS              0
                11              PLANNING                0
                12              SOFTWARE SUPPORT                0
                13              SPIFFY COMPUTER SERVICE DIV.            0
                13              SUPPORT SERVICES                0
                14              SUPPORT SERVICES                0

Column Information:
-------------------
        Number:                                         1
        Name:                                           DEPTNO
        Statistics Available:                           Yes

        Column Statistics:
        ------------------
                Schema name of the column type:                 SYSIBM
                Name of column type:                            CHARACTER
                Maximum column length:                          3
                Scale for decimal or timestamp column:          0
                Number of distinct column values:               14
                Average column length:                          3
                Number of most frequent values:                 -1
                Number of quantiles:                            14
                Second highest data value:                      I22
                Second lowest data value:                       B01
                Column sequence in partition key:               0
                Average number of sub-elements:                 -1
                Average length of delimiters:                   -1
                Percentage encoded column values:               -1
                Column clustering:                              -1

        Column Distribution Statistics:
        -------------------------------
                Quantile Statistics:
                Valcount        Value           Distcount
                ------------------------------------------------
                1               A00             1
                2               B01             2
                3               C01             3
                4               D01             4
                5               D11             5
```

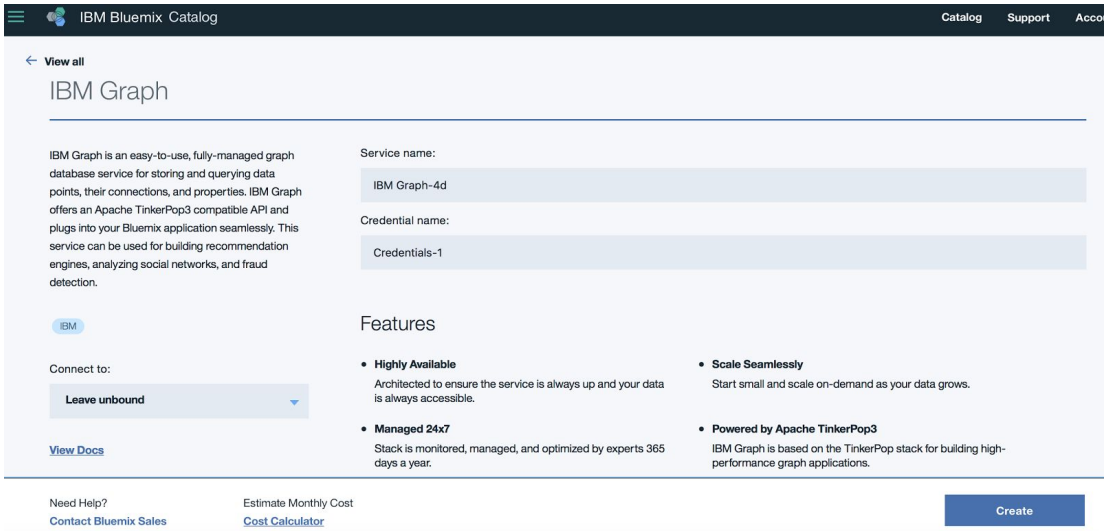
```
                6               D21             6
                7               E01             7
                8               E11             8
                9               E21             9
                10              F22             10
                11              G22             11
                12              H22             12
                13              I22             13
                14              J22             14

Indexes defined on the table:
-----------------------------
Name  : XDEPT3
Schema: DB2INST1
        Unique Rule:                                    Duplicate index
        Used in Operator:                               No
        Page Fetch Pairs:                               Not Available
        Number of Columns:                              1
        Index Leaf Pages:                               1
        Index Tree Levels:                              1
        Index First Key Cardinality:                    3
        Index Full Key Cardinality:                     3
        Index Cluster Ratio:                            100
        Index Cluster Factor:                           -1.000000
        Time of Creation:                               2017-02-28-06.26.06.959282
        Last Statistics Update:                         2017-02-28-06.40.36.588812
        Index Sequential Pages:                         0
        Index First 2 Keys Cardinality:                 -1
        Index First 3 Keys Cardinality:                 -1
        Index First 4 Keys Cardinality:                 -1
        Index Avg Gap between Sequences:                0.000000
        Fetch Avg Gap between Sequences:                -1.000000
        Index Avg Sequential Pages:                     0.000000
        Fetch Avg Sequential Pages:                     -1.000000
        Index Avg Random Pages:                         1.000000
        Fetch Avg Random Pages:                         -1.000000
        Index RID Count:                                14
        Index Deleted RID Count:                        0
        Index Empty Leaf Pages:                         0
        Avg Partition Cluster Ratio:                    -1
        Avg Partition Cluster Factor:                   -1.000000
        Data Partition Cluster Factor:                  1.000000
        Data Partition Page Fetch Pairs:                Not Available

Name  : XDEPT2
Schema: DB2INST1
        Unique Rule:                                    Duplicate index
        Used in Operator:                               No
        Page Fetch Pairs:                               Not Available
        Number of Columns:                              1
        Index Leaf Pages:                               1
        Index Tree Levels:                              1
        Index First Key Cardinality:                    9
        Index Full Key Cardinality:                     9
```

```
        Index Cluster Ratio:                          100
        Index Cluster Factor:                         -1.000000
        Time of Creation:                             2017-02-28-06.26.06.827432
        Last Statistics Update:                       2017-02-28-06.40.36.588812
        Index Sequential Pages:                       0
        Index First 2 Keys Cardinality:               -1
        Index First 3 Keys Cardinality:               -1
        Index First 4 Keys Cardinality:               -1
        Index Avg Gap between Sequences:              0.000000
        Fetch Avg Gap between Sequences:              -1.000000
        Index Avg Sequential Pages:                   0.000000
        Fetch Avg Sequential Pages:                   -1.000000
        Index Avg Random Pages:                       1.000000
        Fetch Avg Random Pages:                       -1.000000
        Index RID Count:                              14
        Index Deleted RID Count:                      0
        Index Empty Leaf Pages:                       0
        Avg Partition Cluster Ratio:                  -1
        Avg Partition Cluster Factor:                 -1.000000
        Data Partition Cluster Factor:                1.000000
        Data Partition Page Fetch Pairs:              Not Available

Name  : PK_DEPARTMENT
Schema: DB2INST1
        Unique Rule:                                  Primary key index
        Used in Operator:                             No
        Page Fetch Pairs:                             Not Available
        Number of Columns:                            1
        Index Leaf Pages:                             1
        Index Tree Levels:                            1
        Index First Key Cardinality:                  14
        Index Full Key Cardinality:                   14
        Index Cluster Ratio:                          100
        Index Cluster Factor:                         -1.000000
        Time of Creation:                             2017-02-28-06.26.05.534125
        Last Statistics Update:                       2017-02-28-06.40.36.588812
        Index Sequential Pages:                       0
        Index First 2 Keys Cardinality:               -1
        Index First 3 Keys Cardinality:               -1
        Index First 4 Keys Cardinality:               -1
        Index Avg Gap between Sequences:              0.000000
        Fetch Avg Gap between Sequences:              -1.000000
        Index Avg Sequential Pages:                   0.000000
        Fetch Avg Sequential Pages:                   -1.000000
        Index Avg Random Pages:                       1.000000
        Fetch Avg Random Pages:                       -1.000000
        Index RID Count:                              14
        Index Deleted RID Count:                      0
        Index Empty Leaf Pages:                       0
        Avg Partition Cluster Ratio:                  -1
        Avg Partition Cluster Factor:                 -1.000000
        Data Partition Cluster Factor:                1.000000
        Data Partition Page Fetch Pairs:              Not Available


Executing Connect Reset -- Connect Reset was Successful.
[db2inst1@9524da63b3cb misc]$
```

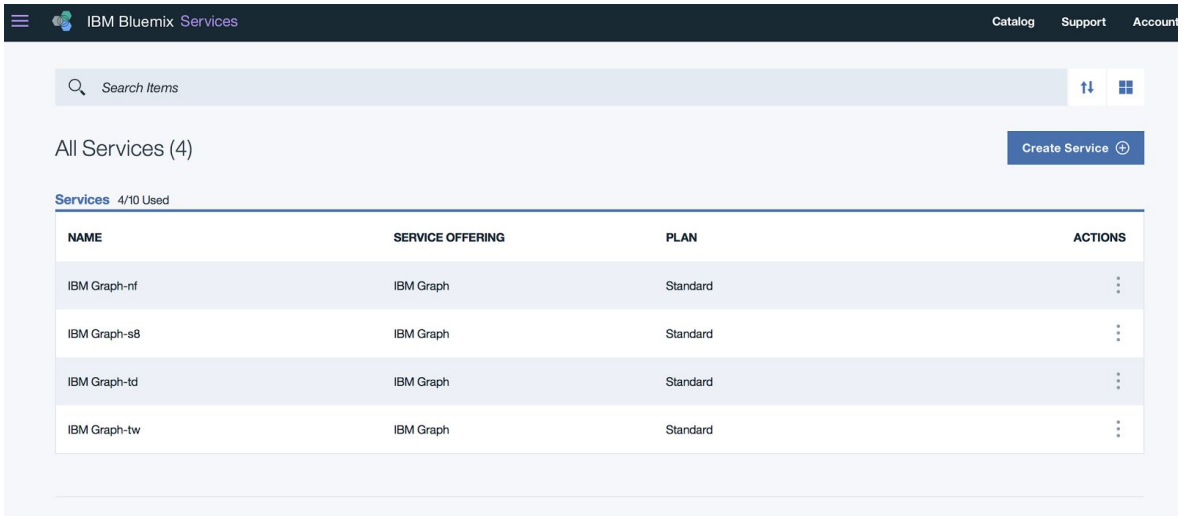
Graph Data store - Bluemix

1. Sign up for IBM Bluemix and login to it
2. Login to it.
3. Click on IBM Graph service, create the service

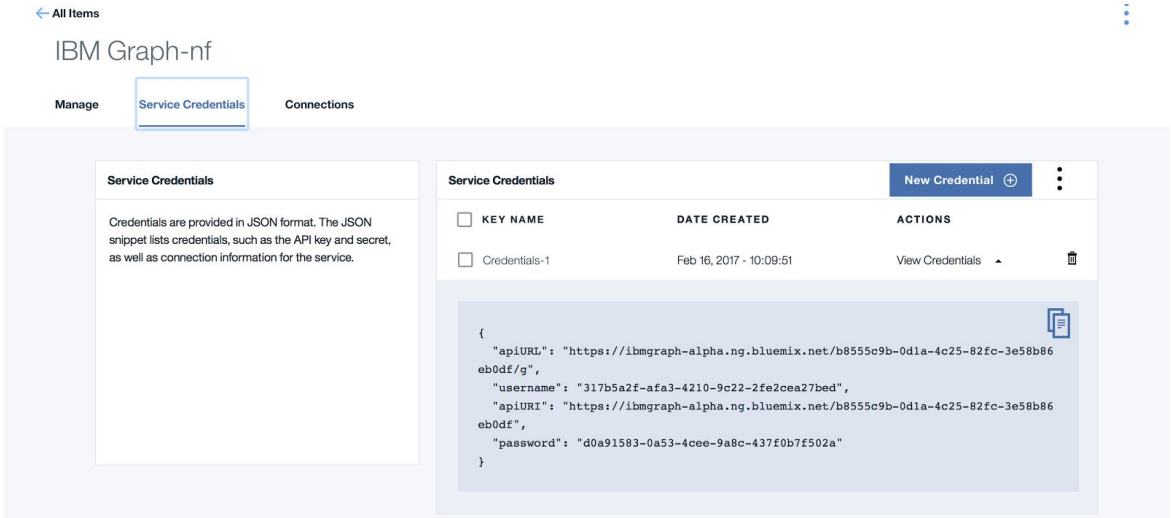




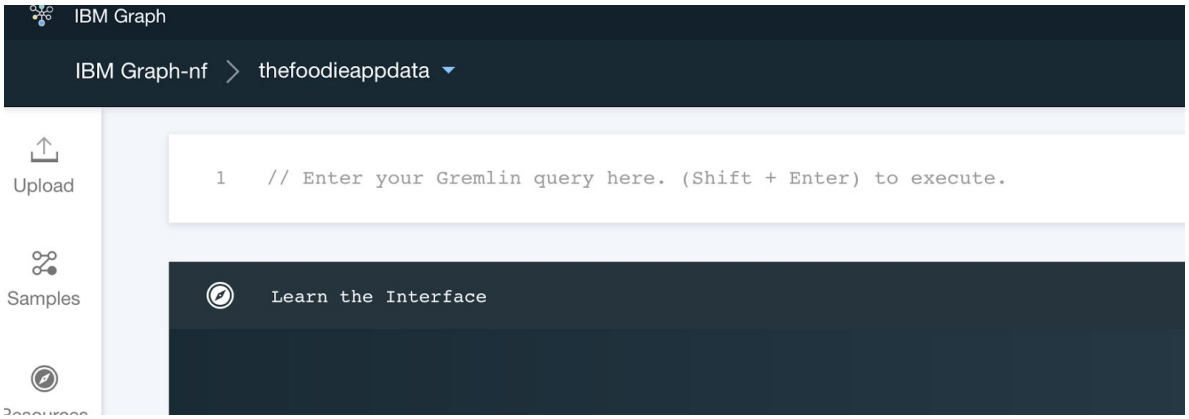
4. Choose a service



5. Make a note of credentials in of the credentials



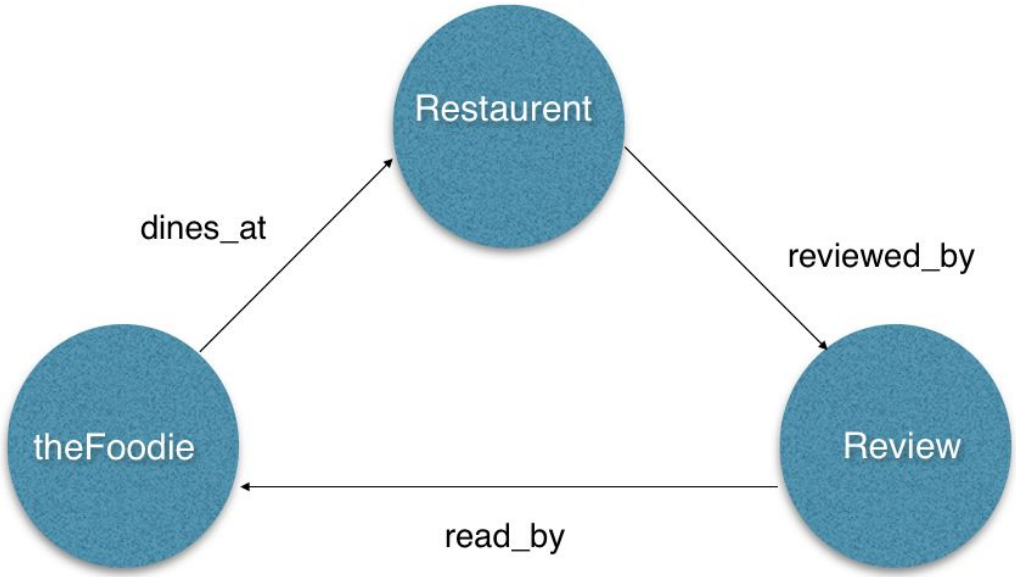
6. Open the service by clicking ‘open’ under ‘Manage’ tab.



7. Create a graph (thefoodieappdata) under the service using curl commands for the terminal

```
Downloads — root@ip-172-31-36-70:/usr/local — -bash — 157×50
Sais-MacBook-Pro:Downloads mulumoodi$ GRAPH="thefoodieappdata"
Sais-MacBook-Pro:Downloads mulumoodi$ curl "$URL/_graphs/$GRAPH" \
>      -X POST \
>      -H "Authorization: gds-token $TOKEN" \
>      -d '' | jq '.'
{
  "graphId": "thefoodieappdata",
  "dbUrl": "https://ibmgraph-alpha.ng.bluemix.net/b8555c9b-0d1a-4c25-82fc-3e58b86eb0df/thefoodieappdata"
}
Sais-MacBook-Pro:Downloads mulumoodi$ curl "$URL/_graphs" \
>      -H "Authorization: gds-token $TOKEN" \
>   | jq -r '.graphs[]' | grep "$GRAPH"
thefoodieappdata
Sais-MacBook-Pro:Downloads mulumoodi$
```

8. Model chosen is as below



9. Upload a schema.

```
Sais-MacBook-Pro:Downloads mulumoodi$ curl "$URL/_graphs" \
>      -H "Authorization: gds-token $TOKEN" \
>   | jq -r '.graphs[]' | grep "$GRAPH"
thefoodieapp-data
Sais-MacBook-Pro:Downloads mulumoodi$ SCHEMA='
> {
>   "propertyKeys": [
>     {"name": "name", "dataType": "String", "cardinality": "SINGLE"},
>     {"name": "review", "dataType": "String", "cardinality": "SINGLE"},
>     {"name": "author", "dataType": "String", "cardinality": "SINGLE"},
>     {"name": "age", "dataType": "Integer", "cardinality": "SINGLE"},
>     {"name": "rating", "dataType": "Integer", "cardinality": "SINGLE"},
>     {"name": "time", "dataType": "String", "cardinality": "SINGLE"}
>   ],
>   "vertexLabels": [
>     {"name": "theFoodie"},
>     {"name": "restaurant"},
>     {"name": "review"}
>   ],
>   "edgeLabels": [
>     {"name": "dines_at", "multiplicity": "MULTI"},
>     {"name": "reviewed_by", "multiplicity": "MULTI"},
>     {"name": "read_by", "multiplicity": "MULTI"}
>   ],
>   "vertexIndexes": [
>     {"name": "vByName", "propertyKeys": ["name"], "composite": true, "unique": true},
>     {"name": "vByReview", "propertyKeys": ["review"], "composite": true, "unique": false},
>     {"name": "vByAuthor", "propertyKeys": ["author"], "composite": true, "unique": false},
>     {"name": "vByAge", "propertyKeys": ["age"], "composite": true, "unique": false},
>     {"name": "vByRating", "propertyKeys": ["rating"], "composite": true, "unique": false}
>   ],
>   "edgeIndexes" :[
>     {"name": "eByTime", "propertyKeys": ["time"], "composite": true, "unique": false}
>   ]
> }'
Sais-MacBook-Pro:Downloads mulumoodi$
Sais-MacBook-Pro:Downloads mulumoodi$ curl "$URL/$GRAPH/schema" \
>      -X POST \
>      -H "Authorization: gds-token $TOKEN" \
>      -H 'Content-Type: application/json' \
>      -d "$SCHEMA" | jq '.'
{
  "requestId": "a4cdb61e-3c08-4d54-b90c-44b3c845cdf3",
  "status": {
    "message": "",
    "code": 200,
    "attributes": {}
  },
  "result": {
    "data": [
      {
        "propertyKeys": [
          {
            "name": "name",
            "dataType": "String",
            "cardinality": "SINGLE"
          },
          {
            "name": "review",
            "dataType": "String",
            "cardinality": "SINGLE"
```

10. Run a set of gremlin queries to update data

```
Downloads — root@ip-172-31-36-70:/usr/local — -bash — 157×50
}
Sais-MacBook-Pro:Downloads mulumoodi$ cat << ENDGREMLIN >gremlin.json
> {
>   "gremlin": "
>   def amanda =  graph.addVertex(T.label, 'theFoodie', 'name', 'Amanda');
>   def anaya =  graph.addVertex(T.label, 'theFoodie', 'name', 'Anaya');
>   def lagartha =  graph.addVertex(T.label, 'theFoodie', 'name', 'Lagartha');
>   def norraine =  graph.addVertex(T.label, 'theFoodie', 'name', 'Norraine');
>
>   def theChinaKing =  graph.addVertex(T.label, 'restaurant', 'name', 'The China King');
>   def italysHeritage =  graph.addVertex(T.label, 'restaurant', 'name', 'Italys Heritage');
>   def indianSpices =  graph.addVertex(T.label, 'restaurant', 'name', 'Indian Spices');
>
>   def rev1 = graph.addVertex(T.label, 'review', 'review', 'ChinaKing is authentic');
>   def rev2 = graph.addVertex(T.label, 'review', 'review', 'Italys heritage is amazing');
>   def rev3 = graph.addVertex(T.label, 'review', 'review', 'I love Indian Spices');
>
>   amanda.addEdge('dines_at', theChinaKing);
>   theChinaKing.addEdge('reviewed_by', rev1);
>
>   anaya.addEdge('dines_at', italysHeritage);
>   italysHeritage.addEdge('reviewed_by', rev2);
>   "
> }
> ENDGREMLIN
Sais-MacBook-Pro:Downloads mulumoodi$
Sais-MacBook-Pro:Downloads mulumoodi$ curl "$URL/$GRAPH/gremlin" \
>      -X POST \
>      -H "Authorization: gds-token $TOKEN" \
>      -H 'Content-Type: application/json' \
>      -d @gremlin.json | jq '.'
{
  "requestId": "d0e57f81-ac0f-456d-bd71-a7a76604d885",
  "status": {
    "message": "",
    "code": 200,
    "attributes": {}
  },
  "result": {
    "data": [
      {
        "id": "2se-3cg-8ph-6i8",
        "label": "reviewed_by",
        "type": "edge",
        "inVLabel": "review",
        "outVLabel": "restaurant",
        "inV": 8432,
        "outV": 4336
      }
    ],
```

11. Run gremlin queries on the graph interface to validate

```
Graph: thefoodieappdata
def gt = graph.traversal();gt.V().hasLabel("theFoodie").has("name", "Amanda").out("dines_at").values("na

1  [
2    "The China King"
3  ]
```

12. Create a Java Console application using APIs to access the graph

```
theFoodieApp
/Library/Java/JavaVirtualMachines/jdk1.8.0_121.jdk/Contents/Home/bin/java ...
Hello Foodies!
Want to know where your fellow foodie dines?! Pls enter your fauvorite foodie's name
Amanda
"Amanda"goes to The China King
Want to know the review of your fauvorite restaurent! Enter the name of a restaurent
The China King
The review on "The China King" says that "ChinaKing is authentic"

Process finished with exit code 0
```

```
theFoodieApp
/Library/Java/JavaVirtualMachines/jdk1.8.0_121.jdk/Contents/Home/bin/java ...
Hello Foodies!
Want to know where your fellow foodie dines?! Pls enter your fauvorite foodie's name
Anaya
"Anaya"goes to Italys Heritage
Want to know the review of your fauvorite restaurent! Enter the name of a restaurent
Italys Heritage
The review on "Italys Heritage" says that "Italys heritage is amazing"

Process finished with exit code 0
```

13. Snapshots of the Java code

theFoodieApp

```java
import org.apache.commons.codec.binary.Base64;
import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.entity.ContentType;
import org.apache.http.entity.StringEntity;
import org.apache.http.impl.client.HttpClientBuilder;
import org.apache.http.util.EntityUtils;
import org.apache.wink.json4j.JSONArray;
import org.apache.wink.json4j.JSONException;
import org.apache.wink.json4j.JSONObject;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

/**
 * Created by mulumoodi on 2/22/17.
 */
public class theFoodieApp {
    public static void main(String[] args) throws JSONException {
        System.out.println("Hello Foodies!"); // Display the string.
        String apiURL = null;
        String username = null;
        String password = null;
        String baseURL = null;
        String basicAuth = null;

        apiURL= "https://ibmgraph-alpha.ng.bluemix.net/b8555c9b-0d1a-4c25-82fc-3e58b86eb0df/thefoodieappdata";
        username= "317b5a2f-afa3-4210-9c22-2fe2cea27bed";
        baseURL="https://ibmgraph-alpha.ng.bluemix.net/b8555c9b-0d1a-4c25-82fc-3e58b86eb0df";
        password= "d0a91583-0a53-4cee-9a8c-437f0b7f502a";
        byte[] userpass = (username + ":" + password).getBytes();
        byte[] encoding = Base64.encodeBase64(userpass);
        basicAuth = "Basic " + new String(encoding);

        String gdsToken;
        String gdsTokenAuth = null;
        HttpGet httpGet = new HttpGet( uri: baseURL + "/_session");
        httpGet.setHeader( name: "Authorization", basicAuth);
```

```java
        HttpClient client = HttpClientBuilder.create().build();
        HttpResponse httpResponse = null;
        try {
            httpResponse = client.execute(httpGet);
        } catch (IOException e) {
            e.printStackTrace();
        }
        HttpEntity httpEntity = httpResponse.getEntity();
        String content = null;
        try {
            content = EntityUtils.toString(httpEntity);
        } catch (IOException e) {
            e.printStackTrace();
        }
        try {
            EntityUtils.consume(httpEntity);
        } catch (IOException e) {
            e.printStackTrace();
        }
        JSONObject jsonContent = new JSONObject(content);
        gdsToken = jsonContent.getString( key: "gds-token");
        gdsTokenAuth = "gds-token " + gdsToken;
        //System.out.println(gdsTokenAuth);

        System.out.println("Want to know where your fellow foodie dines?! Pls enter your fauvorite foodie's name ");
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        String s = null;
        try {
            s = br.readLine();
            s="\""+s+"\"";
        } catch (IOException e) {
            e.printStackTrace();
        }

        //Running gremlin queries

        JSONObject postData = new JSONObject();
        postData.put("gremlin", "def g = graph.traversal(); g.V().hasLabel(\"theFoodie\").has(\"name\", "+s+ ").out(\"dines_at\").values(\"name\");");
        HttpPost httpPost = new HttpPost( uri: apiURL + "/gremlin");
        httpPost.setHeader( name: "Authorization", gdsTokenAuth);
        StringEntity strEnt = new StringEntity(postData.toString(), ContentType.APPLICATION_JSON);
```

```java
        httpPost.setHeader( name: "Authorization", gdsTokenAuth);
        StringEntity strEnt = new StringEntity(postData.toString(), ContentType.APPLICATION_JSON);
        httpPost.setEntity(strEnt);
        try {
        try {
            httpResponse = client.execute(httpPost);
        } catch (IOException e) {
            e.printStackTrace();
        }
        httpEntity = httpResponse.getEntity();
        try {
            content = EntityUtils.toString(httpEntity);
        } catch (IOException e) {
            e.printStackTrace();
        }
        try {
            EntityUtils.consume(httpEntity);
        } catch (IOException e) {
            e.printStackTrace();
        }
        //System.out.println("Content is "+content);
        jsonContent = new JSONObject(content);
        JSONObject result = jsonContent.getJSONObject("result");
        JSONArray data = result.getJSONArray( key: "data");
        System.out.println(s+"goes to "+data.getString( index: 0));

        //Part 2
        System.out.println("Want to know the review of your fauvorite restaurent! Enter the name of a restaurent ");
        br = new BufferedReader(new InputStreamReader(System.in));
        s = null;
        try {
            s = br.readLine();
            s="\""+s+"\"";
        } catch (IOException e) {
            e.printStackTrace();
        }
        postData = new JSONObject();
        postData.put("gremlin", "def g = graph.traversal(); g.V().hasLabel(\"restaurant\").has(\"name\", "+s+ ").out(\"reviewed_by\").values(\"review\");");
        br = new BufferedReader(new InputStreamReader(System.in));
        s = null;
        try {
            s = br.readLine();
            s="\""+s+"\"";
        } catch (IOException e) {
            e.printStackTrace();
        }
        postData = new JSONObject();
        postData.put("gremlin", "def g = graph.traversal(); g.V().hasLabel(\"restaurant\").has(\"name\", "+s+ ").out(\"reviewed_by\").values(\"review\");");
        httpPost = new HttpPost( url: apiURL + "/gremlin");
        httpPost.setHeader( name: "Authorization", gdsTokenAuth);
        strEnt = new StringEntity(postData.toString(), ContentType.APPLICATION_JSON);
        httpPost.setEntity(strEnt);
        try {
            httpResponse = client.execute(httpPost);
        } catch (IOException e) {
            e.printStackTrace();
        }
        httpEntity = httpResponse.getEntity();
        try {
            content = EntityUtils.toString(httpEntity);
        } catch (IOException e) {
            e.printStackTrace();
        }
        try {
            EntityUtils.consume(httpEntity);
        } catch (IOException e) {
            e.printStackTrace();
        }
        //System.out.println("Content is "+content);
        jsonContent = new JSONObject(content);
        result = jsonContent.getJSONObject("result");
        data = result.getJSONArray( key: "data");
        System.out.println("The review on "+s+" says that \""+data.getString( index: 0)+"\"");

    }
}
```