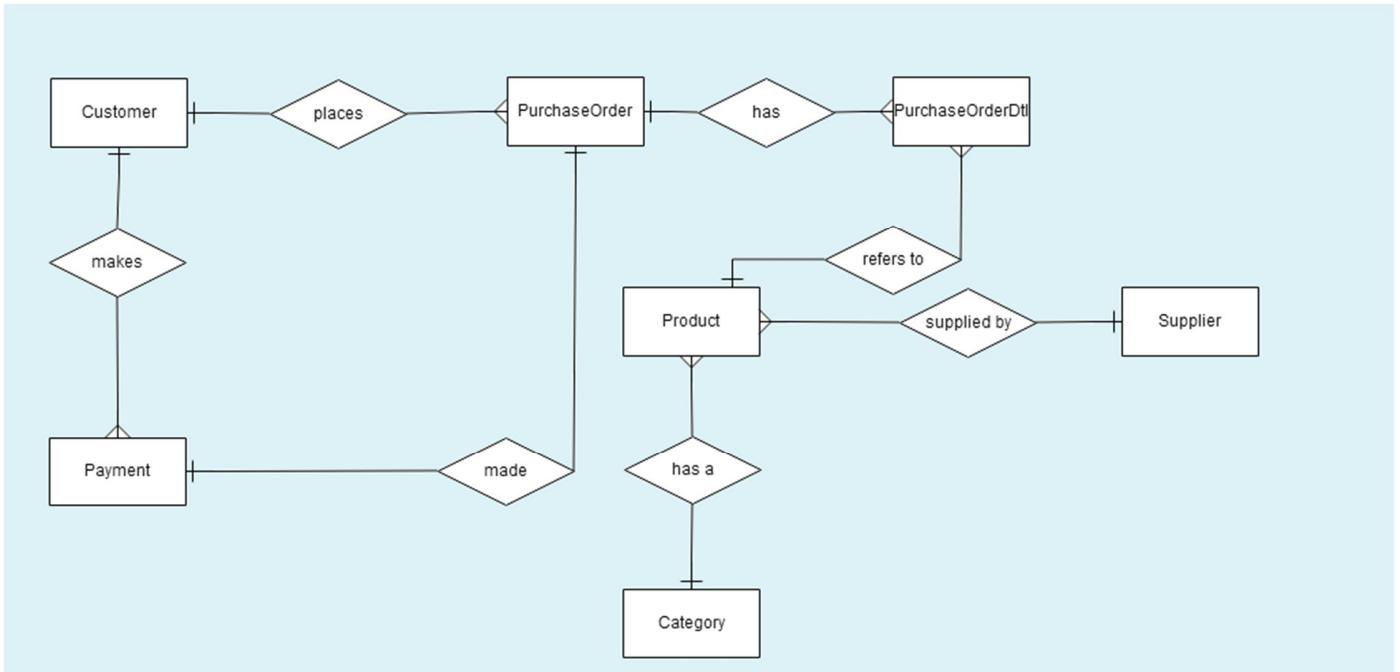


Database Team Based Assignment (Team 19)

1. SQLite Assignment

Database Design For Purchase Order Management System (ER -Diagram):



Sample Schema with tables

Structure | Browse & Search | Execute SQL | DB Settings

TABLE: Product

Drop Empty Rename Reindex Copy Export

Create statement

```
CREATE TABLE "Product" ("product_id" VARCHAR PRIMARY KEY NOT NULL, "product_name" VARCHAR NOT NULL, "category_id" INTEGER NOT NULL, "unit_price" FLOAT NOT NULL, "supplier_id" VARCHAR NOT NULL, "units_in_stock" INTEGER NOT NULL, "last_update_dt" DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP)
```

More Info

No. of Records: 7 No. of Indexes: 1 No. of Triggers: 0

Columns (7)

Column ID	Name	Type	Not Null	Default Value	Primary Key
0	product_id	VARCHAR	1	null	1
1	product_name	VARCHAR	1	null	0
2	category_id	INTEGER	1	null	0
3	unit_price	FLOAT	1	null	0
4	supplier_id	VARCHAR	1	null	0
5	units_in_stock	INTEGER	1	null	0
6	last_update_dt	DATETIME	1	CURRENT_TIMESTAMP	0

TABLE: PurchaseOrder

[Drop](#) [Empty](#) [Rename](#) [Reindex](#) [Copy](#) [Export](#)

Create statement

```
CREATE TABLE PurchaseOrder (
    order_id VARCHAR, cust_id INTEGER NOT NULL,
    po_create_dt DATETIME NOT NULL , po_amount FLOAT NOT NULL , po_status VARCHAR NOT NULL,
    PRIMARY KEY (order_id)
    FOREIGN KEY (cust_id) references Customer(cust_id) ON DELETE CASCADE
)
```

More Info

No. of Records: 3 No. of Indexes: 1 No. of Triggers: 0

Columns (5)

Column ID	Name	Type	Not Null	Default Value	Primary Key
0	order_id	VARCHAR	0	null	1
1	cust_id	INTEGER	1	null	0
2	po_create_dt	DATETIME	1	null	0
3	po_amount	FLOAT	1	null	0
4	po_status	VARCHAR	1	null	0

TABLE: PurchaseOrdDtl

[Drop](#) [Empty](#) [Rename](#) [Reindex](#) [Copy](#) [Export](#)

Create statement

```
CREATE TABLE "PurchaseOrdDtl" ("order_id" VARCHAR NOT NULL ,
"product_id" INTEGER,
"Quantity" INTEGER NOT NULL ,
"last_update_dt" DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
PRIMARY KEY (order_id,product_id),
FOREIGN KEY (order_id) references PurchaseOrdr(order_id) ON DELETE CASCADE
FOREIGN KEY (product_id) references Product(product_id) ON DELETE SET NULL)
```

More Info

No. of Records: 6 No. of Indexes: 1 No. of Triggers: 0

Columns (4)

Column ID	Name	Type	Not Null	Default Value	Primary Key
0	order_id	VARCHAR	1	null	1
1	product_id	INTEGER	0	null	2
2	Quantity	INTEGER	1	null	0
3	last_update_dt	DATETIME	1	CURRENT_TIMESTAMP	0

[Structure](#) [Browse & Search](#) [Execute SQL](#) [DB Settings](#)

TABLE: Category

[Drop](#) [Empty](#) [Rename](#) [Reindex](#) [Copy](#) [Export](#)

Create statement

```
CREATE TABLE "Category" ("category_id" VARCHAR PRIMARY KEY NOT NULL , "category_name" VARCHAR NOT NULL , "last_update_dt" DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP)
```

More Info

No. of Records: 4 No. of Indexes: 1 No. of Triggers: 0

Columns (3)

Column ID	Name	Type	Not Null	Default Value	Primary Key
0	category_id	VARCHAR	1	null	1
1	category_name	VARCHAR	1	null	0
2	last_update_dt	DATETIME	1	CURRENT_TIMESTAMP	0

Structure | Browse & Search | Execute SQL | DB Settings

TABLE: Customer

Create statement

```
CREATE TABLE "Customer" ("cust_id" VARCHAR PRIMARY KEY NOT NULL , "cust_name" VARCHAR NOT NULL , "cust_address" VARCHAR NOT NULL , "cust_phone" VARCHAR, "cust_email" VARCHAR, "cust_create_dt" DATETIME NOT NULL )
```

More Info

No. of Records: 3 No. of Indexes: 1 No. of Triggers: 0

Columns (6)

Column ID	Name	Type	Not Null	Default Value	Primary Key
0	cust_id	VARCHAR	1	null	1
1	cust_name	VARCHAR	1	null	0
2	cust_address	VARCHAR	1	null	0
3	cust_phone	VARCHAR	0	null	0
4	cust_email	VARCHAR	0	null	0
5	cust_create_dt	DATETIME	1	null	0

Structure | Browse & Search | Execute SQL | DB Settings

TABLE: Supplier

Create statement

```
CREATE TABLE "Supplier" ("supplier_id" VARCHAR PRIMARY KEY NOT NULL , "supplier_name" VARCHAR NOT NULL , "supplier_address" VARCHAR NOT NULL , "supplier_phone" VARCHAR NOT NULL , "supplier_email" VARCHAR, "supplier_create_dt" DATETIME NOT NULL )
```

More Info

No. of Records: 5 No. of Indexes: 1 No. of Triggers: 0

Columns (6)

Column ID	Name	Type	Not Null	Default Value	Primary Key
0	supplier_id	VARCHAR	1	null	1
1	supplier_name	VARCHAR	1	null	0
2	supplier_address	VARCHAR	1	null	0
3	supplier_phone	VARCHAR	1	null	0
4	supplier_email	VARCHAR	0	null	0
5	supplier_create_dt	DATETIME	1	null	0

Structure | Browse & Search | Execute SQL | DB Settings

TABLE: Payment

Create statement

```
CREATE TABLE "Payment" ("cust_id" INTEGER NOT NULL , "check_num" VARCHAR NOT NULL , "payment_dt" DATETIME NOT NULL , "order_id" VARCHAR NOT NULL , "amount" FLOAT NOT NULL , "status" VARCHAR NOT NULL , PRIMARY KEY (cust_id, order_id) FOREIGN KEY (order_id) references PurchaseOrdr(order_id) ON DELETE RESTRICT )
```

More Info

No. of Records: 3 No. of Indexes: 1 No. of Triggers: 0

Columns (6)

Column ID	Name	Type	Not Null	Default Value	Primary Key
0	cust_id	INTEGER	1	null	1
1	check_num	VARCHAR	1	null	0
2	payment_dt	DATETIME	1	null	0
3	order_id	VARCHAR	1	null	2
4	amount	FLOAT	1	null	0
5	status	VARCHAR	1	null	0

Sample Data:

TABLE Product							
	product_id	product_name	category_id	unit_price	supplier_id	units_in_stock	last_update_dt
1	100	Oreo mini cookies	1	4.99	G121	50	2017-02-21 09:56:27
2	101	Nutty bars	1	3.72	G121	35	2017-02-21 09:57:18
3	102	Pringles BBQ	1	3.72	G142	100	2017-02-21 09:59:05
4	103	Crayola colored ...	2	1.84	S200	70	2017-02-21 10:02:08
5	104	Viper nano drone	4	25.99	E210	15	2017-02-21 10:04:57
6	105	MG Desk lamp	3	38.69	S200	30	2017-02-21 10:06:39
7	106	WP mini refriger...	3	48.55	H144	10	2017-02-21 10:07:45

TABLE PurchaseOrder					
	order_id	cust_id	po_create_dt	po_amount	po_status
1	P2000	1001	2017-02-21 19:26:15	25.99	Created
2	P2001	1001	2017-02-21 19:27:56	63.43	In Progress
3	P2002	1003	2017-02-21 19:29:11	35.32	Closed

TABLE PurchaseOrdDtl					
	order_id	product_id	Quantity	last_update_dt	
1	P2000	104	1	2017-02-21 20:10:04	
2	P2001	106	1	2017-02-21 20:10:29	
3	P2001	102	4	2017-02-21 20:11:10	
4	P2002	101	4	2017-02-21 20:12:41	
5	P2002	102	5	2017-02-21 20:12:56	
6	P2002	103	1	2017-02-21 20:13:09	

TABLE Category					
	category_id	category_name	last_update_dt		
1	1	Grocery	2017-02-21 19:18:59		
2	2	Stationary	2017-02-21 19:19:13		
3	3	Home Supplies	2017-02-21 19:19:27		
4	4	Electronics	2017-02-21 19:19:40		

TABLE Customer						
	cust_id	cust_name	cust_address	cust_phone	cust_email	cust_create_dt
1	1000	Dana	456, xy street, CA	1237689956	dana@gmail.com	2017-02-21 19:16:54
2	1001	Smith	134, 7th street, CA	8906784537	smith@gmail.com	2017-02-21 19:17:54
3	1002	Betty	121, ab street, TX	5673249088	betty@ab.com	2017-02-21 19:18:38

TABLE Supplier						
	supplier_id	supplier_name	supplier_address	supplier_phone	supplier_email	supplier_create_dt
1	E210	Viper	789, abc street, CO	1234789682	viper@abc.com	2017-02-21 19:08:54
2	G142	Jack trading	675, union city, NJ	7481117839	jt@gmail.com	2017-02-21 19:11:24
3	G121	Debbie	444, first street, CA	4578905678	debbie@abc.com	2017-02-21 19:14:04
4	S200	MG	444, Third street, TX	5672349008	mg@abc.com	2017-02-21 19:14:52
5	H144	Whirlpool	444, bg street, NY	2347859812	whlpool@abc.com	2017-02-21 19:15:33

TABLE Payment						
	check_num	payment_dt	order_id	amount	status	
1001	C4689653	2017-02-21 23:52:49	P2000	25.99	success	
1001	C4689653	2017-02-21 23:53:25	P2001	63.43	success	
1003	C45789953	2017-02-21 23:54:06	P2002	35.32	success	

Queries:

1) SELECT order_id, SUM(po_amount) AS 'Total Amount', cust_id FROM PurchaseOrder WHERE po_status IN ('Created','In Progress') GROUP BY cust_id

Enter SQL

```
SELECT order_id, SUM(po_amount) AS 'Total Amount', cust_id FROM PurchaseOrder WHERE po_status IN ('Created','In Progress') GROUP BY cust_id
```

Run SQL Actions Last Error: not an error

order_id	Total Amount	cust_id
P2001	89.42	1001
P1006	149	1002
P1004	120.99	1003
P1007	35.69	1004

2) SELECT p.order_id, p.cust_id, check_num, amount, status FROM PurchaseOrder AS p, Payment WHERE p.order_id = payment.order_id AND p.cust_id = Payment.cust_id ORDER BY p.order_id DESC

```
SELECT p.order_id, p.cust_id, check_num, amount, status FROM PurchaseOrder AS p, Payment WHERE p.order_id = payment.order_id AND p.cust_id = Payment.cust_id ORDER BY p.order_id DESC
```

Run SQL Actions Last Error: not an error

order_id	cust_id	check_num	amount	status
P2002	1003	C45789953	35.32	success
P2001	1001	C4689653	63.43	success
P2000	1001	C4689653	25.99	success

3) SELECT p.product_id,product_name, c.category_id,c.category_name FROM Product AS p LEFT JOIN Category AS c WHERE p.category_id = c.category_id

Enter SQL

```
SELECT p.product_id,product_name, c.category_id,c.category_name FROM Product AS p LEFT JOIN Category AS c WHERE p.category_id = c.category_id
```

Run SQL Actions Last Error: not an error

product_id	product_name	category_id	category_name
100	Oreo mini cookies	1	Grocery
101	Nutty bars	1	Grocery
102	Pringles BBQ	1	Grocery
103	Crayola colored pencils	2	Stationary
104	Viper nano drone	4	Electronics
105	MG Desk lamp	3	Home Supplies
106	WP mini refrigerator	3	Home Supplies

4) DELETE FROM Customer WHERE cust_id = 1001
(On Delete Cascade)

TABLE Customer

Customer							Search	Show All	Add	Duplicate	Edit	Delete
rowid	cust_id	cust_name	cust_address	cust_phone	cust_email	cust_create_dt						
1	1000	Dana	456, xy street, CA	1237689956	dana@gmail.com	2017-02-21 19:16:54						
3	1002	Betty	121, ab street, TX	5673249088	betty@ab.com	2017-02-21 19:18:38						

TABLE PurchaseOrder		Search	Show All	Add	Duplicate	Edit	Delete
rowid	order_id	cust_id	po_create_dt	po_amount	po_status		
3	P2002	1003	2017-02-22 08:51:07	35.32	In Progress		

5) DELETE FROM Product WHERE product_id = 101

(On Delete Set NULL)

Structure								Browse & Search	Execute SQL	DB Settings	
TABLE Product		Search	Show All	Add	Duplicate	Edit	Delete				
rowid	product_id	product_name	category_id	unit_price	supplier_id	units_in_stock	last_update...				
1	100	Oreo mini cook...	1	4.99	G121	50	2017-02-21 09:56:2				
3	102	Pringles BBQ	1	3.72	G142	100	2017-02-21 09:59:0				
4	103	Crayola colored...	2	1.84	S200	70	2017-02-21 10:02:0				
5	104	Viper nano drone	4	25.99	E210	15	2017-02-21 10:04:5				
6	105	MG Desk lamp	3	38.69	S200	30	2017-02-21 10:06:3				
7	106	WP mini refrige...	3	48.55	H144	10	2017-02-21 10:07:4				

TABLE PurchaseOrdDtl		Search	Show All	Add	Duplicate	Edit	Delete
rowid	order_id	product_id	Quantity	last_update_dt			
1	P2000	104	1	2017-02-22 08:52:03			
2	P2001	106	1	2017-02-22 08:52:16			
3	P2001	102	4	2017-02-22 08:52:26			
4	P2002		4	2017-02-22 08:52:40			
5	P2002	102	5	2017-02-22 08:52:54			
6	P2002	103	1	2017-02-22 08:53:03			

2. DB2 Assignment

Query using where and group by conditions:

```
DB2 CLP - DB2COPY1 - C:\PROGRA~1\IBM\SQLLIB\BIN\db2setcp.bat DB2SETCP.BAT DB2.EXE
db2 => connect to sampldb2
      Database Connection Information
      Database server      = DB2/NT64 11.1.1.1
      SQL authorization ID = XINFERNA...
      Local database alias  = SAMPLDB2

db2 => select P.paymentid,C.customerid,C.customername,P.amount from customer C, payment P where C.customerid = P.customerid and P.amount > 100
PAYMENTID CUSTOMERID CUSTOMERNAME          AMOUNT
-----  -----  -----
1003     1       Alex                305.66
1002     1       Alex                282.78
1001     1       Alex                150.2
1034     2       Justin               456.98
1078     2       Justin               101.78
1089     3       Ricky                178.98
1478     4       Steve                111.00

7 record(s) selected.

db2 => select P.customerid,C.customername,sum(P.amount) as Total_Amount from customer C, payment P where C.customerid = P.customerid group by P.customerid,C.customername
CUSTOMERID CUSTOMERNAME          TOTAL_AMOUNT
-----  -----
1       Alex                  697.76
2       Justin                 641.01
3       Ricky                  257.80
4       Steve                  154.45
5       Angel                  155.46

5 record(s) selected.

db2 => -
```

Generate query explain plan:

Query :- Select P.customerid, C.customername, sum(P.amount) as Total Amount from customer
C, payment P where C.customerid = P.customerid group by P.customerid, C.customername;

1) Creating Explain tables

- 2) a) Setting Explain mode to 'Explain'
- b) Executing the query in Explain mode
- c) Setting Explain mode to 'No'

```
C:\> Administrator:DB2 CLP - DB2COPY1
C:\>Program Files\IBM\SQLLIB\BIN>db2 set current explain mode explain
DB20000I: The SQL command completed successfully.

C:\>Program Files\IBM\SQLLIB\BIN>db2 select P.customerid,C.customername,sum(P.amount) as Total_Amount from customer C, payment P where C.customerid = P.customerid group by P.customerid,C.customername
SQL0217W: The statement was not executed as only Explain information requests are being processed. SQLSTATE=01004
C:\>Program Files\IBM\SQLLIB\BIN>db2 set current explain mode no
DB20000I: The SQL command completed successfully.

C:\>Program Files\IBM\SQLLIB\BIN>
```

- 3) Generating query explain plan using db2exfmt tool

```
C:\> Administrator:DB2 CLP - DB2COPY1
C:\>Program Files\IBM\SQLLIB\BIN>db2exfmt
DB20000I: The SQL command completed successfully.

C:\>Program Files\IBM\SQLLIB\BIN>DB2 Universal Database Version 11.1, 5622-044 (c) Copyright IBM Corp. 1991, 2015
Licensed Material - Program Property of IBM
IBM DATABASE 2 Explain Table Format Tool

Enter Database Name --> sampdb2
Connecting to the database...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 connection name for the Infernal tool -->
Enter outfile name. Default is to terminal -->
DB2 Universal Database Version 11.1, 5622-044 (c) Copyright IBM Corp. 1991, 2015
Licensed Material - Program Property of IBM
IBM DATABASE 2 Explain Table Format Tool

***** EXPLAIN INSTANCE *****

DB2 VERSION: 11.01.4
FORMATTED ON DB: SAMPDB2
SOURCE_NAME: SQL2002
SOURCE_SCHEMA: NULLID
EXPLAIN_TIMESTAMP: 2017-02-21-20.41.05.704000
EXPLAIN_TIME: 2017-02-21-20.41.05.704000
EXPLAIN_REQUESTER: XINFERNAL

Database context:
-----
Parallelism: None
CPU Speed: 7.872404e-008
Comm Speed: 0
Buffer Pool size: 250
Sort Heap size: 250
Database Log size: 400
Lock List size: 4000
Maximum Lock List: 22
Average Applications: 1
Locks Available: 28835

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

----- STATEMENT 1 SECTION 201 -----
QUERYNO: 9
QUERYTEXT: CLP
Statement Type: Select
Updatable: No
Deletable: No
Query Degree: 1

Original Statement:
```

- Optimized Query

```

Administrator:DB2 CLP - DB2COPY1
-----+-----+-----+-----+-----+
-----| Statement 1 SECTION 201 |
-----+-----+-----+-----+-----+
  QUERYNO:          9
  QUERYTYPE:        CLP
  Statement Type:   Select
  Updatable:        No
  Deletable:         No
  Query Degree:    1

Original Statement:
select
  P.customerid,
  C.customername,
  sum(P.amount) as Total_Amount
from
  customer C,
  payment P
where
  C.customerid = P.customerid
group by
  P.customerid,
  C.customername

Optimized Statement:
-----+-----+-----+-----+-----+
-----| SELECT
  Q4.CUSTOMERID AS "CUSTOMERID",
  Q4.CUSTOMERNAME AS "CUSTOMERNAME",
  Q4.$C2 AS "TOTAL_AMOUNT"
FROM
  (SELECT
    Q2.CUSTOMERID,
    Q2.CUSTOMERNAME,
    SUM(DECFL0AT(Q3.AMOUNT, 34, '-'))
  FROM
    (SELECT
      Q1.CUSTOMERID,
      Q1.CUSTOMERNAME,
      Q1.AMOUNT
    FROM
      XINFERAL_PAYMENT AS Q1,
      XINFERAL_CUSTOMER AS Q2
    WHERE
      (Q2.CUSTOMERID = Q1.CUSTOMERID)
    ) AS Q3
  GROUP BY
    Q3.CUSTOMERNAME,
    Q3.CUSTOMERID
  ) AS Q4
-----+-----+-----+-----+-----+
Access Plan:
-----+-----+-----+-----+-----+
-----| Total Cost:       13.6228
  | Query Degree:    1
  |
  | Rows           RETURN
  | ( 1)           ( 1)
  | Cost           Cost
  | I/O            I/O
  |                |
  |                5
-----+-----+-----+-----+-----+

```

- Access Plan

```

Administrator:DB2 CLP - DB2COPY1
-----+-----+-----+-----+-----+
-----| AS Q3
  | GROUP BY
  | Q3.CUSTOMERNAME,
  | Q3.CUSTOMERID
  | ) AS Q4
Access Plan:
-----+-----+-----+-----+-----+
-----| Total Cost:       13.6228
  | Query Degree:    1
  |
  | Rows           RETURN
  | ( 1)           ( 1)
  | Cost           Cost
  | I/O            I/O
  |                |
  |                5
  | GROUP BY
  | ( 2)           ( 2)
  | 13.6226
  | 2
  |                |
  | JOIN           HSJOIN
  | ( 3)           ( 3)
  | 13.6223
  | 2
  |                |
  |-----+-----|
  | 5   TBSCAN     FILTER
  | ( 4)   ( / )
  | 6.81036   6.81128
  | 1
  | 6   12
  |                |
  | SORT           TBSCAN
  | ( 5)   ( 8)
  | 6.81026   6.81128
  | 1
  | 6   12
  |                |
  | TBSCAN     SORT
  | ( 6)   ( 9)
  | 6.80993   6.81118
  | 1
  | 6   12
  | TABLE: XINFERAL TBSCAN
  | CUSTOMER   ( 10)
  | Q2        6.81074
  | 1
  | 12
  | TABLE: XINFERAL PAYMENT
  | Q1
-----+-----+-----+-----+-----+
Extended Diagnostic Information:
-----+-----+-----+-----+-----+

```

- Plan Details

a) Return Result

```
Administrator: DB2 CLP - DB2COPY1
12
TABLE: S1612051900
PAYMENT
Q1

Extended Diagnostic Information:
-----
No extended Diagnostic Information for this statement.

Plan Details:
-----
1) RETURN: (Return Result)
    Cumulative Total Cost:      13.6228
    Cumulative CPU Cost:       163095
    Cumulative I/O Cost:        2
    Cumulative Re-Total Cost:   13.6228
    Cumulative Re-CPU Cost:    163095
    Cumulative Re-I/O Cost:    2
    Cumulative First Row Cost: 13.6219
    Estimated Bufferpool Buffers: 0

    Arguments:
    -----
    BLDLEVEL: (Build Level)
    DB2_V111101010100 : S1612051900
    HEAPUSE : (Maximum Statement Heap Usage)
    90 Pages
    PLANID : (Access plan identifier)
    7934442c3e3d
    PREPTIME: (Statement prepare time)
    298 milliseconds
    SEMEVID : (Semaphore identifier)
    d2ad5727cfdd1e
    STMTHAP: (Statement heap size)
    8192
    STMTID : (Normalized statement identifier)
    43805d82795579d0

    Input Streams:
    -----
    1) From Operator #2
        Estimated number of rows:      5
        Number of columns:            3
        Subquery predicate ID:        Not Applicable
        Column Names:
        -----
        +Q5.TOTAL_AMOUNT+Q5.CUSTOMERNAME+Q5.CUSTOMERID

2) GRPBY : (Group By)
    Cumulative Total Cost:      13.6226
    Cumulative CPU Cost:       168425
    Cumulative I/O Cost:        2
    Cumulative Re-Total Cost:   13.6226
    Cumulative Re-CPU Cost:    168425
    Cumulative Re-I/O Cost:    2
    Estimated Bufferpool Buffers: 0

    Arguments:
    -----
    AGGMODE : (Aggregation Mode)
    COMPLETE
    GROUPBYC: (Group By columns)
    TRUE
    GROUPBYN: (Number of Group By columns)
    2
    GROUPBYR: (Group By requirement)
    1: Q3.CUSTOMERID
    2: Q3.CUSTOMERNAME
    ONEFETCH: (One Fetch flag)
    FALSE

    Input Streams:
    -----
    1b) From Operator #3
        Estimated number of rows:      12
        Number of columns:            3
        Subquery predicate ID:        Not Applicable
        Column Names:
        -----
        +Q3.CUSTOMERID(A)+Q3.CUSTOMERNAME(A)+Q3.AMOUNT

    Output Streams:
    -----
    1) To Operator #1
        Estimated number of rows:      5
        Number of columns:            3
        Subquery predicate ID:        Not Applicable
        Column Names:
        -----
        +Q5.TOTAL_AMOUNT+Q5.CUSTOMERNAME+Q5.CUSTOMERID

2) MSJOIN: (Merge Scan Join)
    Cumulative Total Cost:      13.6223
    Cumulative CPU Cost:       156871
    Cumulative I/O Cost:        2
    Cumulative Re-Total Cost:   13.6223
```

b) Group By

```
Administrator: DB2 CLP - DB2COPY1
Estimated number of rows:      5
Number of columns:            3
Subquery predicate ID:        Not Applicable

Column Names:
-----
+Q5.TOTAL_AMOUNT+Q5.CUSTOMERNAME+Q5.CUSTOMERID

2) GRPBY : (Group By)
    Cumulative Total Cost:      13.6226
    Cumulative CPU Cost:       168425
    Cumulative I/O Cost:        2
    Cumulative Re-Total Cost:   13.6226
    Cumulative Re-CPU Cost:    168425
    Cumulative Re-I/O Cost:    2
    Cumulative First Row Cost: 13.6219
    Estimated Bufferpool Buffers: 0

    Arguments:
    -----
    AGGMODE : (Aggregation Mode)
    COMPLETE
    GROUPBYC: (Group By columns)
    TRUE
    GROUPBYN: (Number of Group By columns)
    2
    GROUPBYR: (Group By requirement)
    1: Q3.CUSTOMERID
    2: Q3.CUSTOMERNAME
    ONEFETCH: (One Fetch flag)
    FALSE

    Input Streams:
    -----
    1b) From Operator #3
        Estimated number of rows:      12
        Number of columns:            3
        Subquery predicate ID:        Not Applicable
        Column Names:
        -----
        +Q3.CUSTOMERID(A)+Q3.CUSTOMERNAME(A)+Q3.AMOUNT

    Output Streams:
    -----
    1) To Operator #1
        Estimated number of rows:      5
        Number of columns:            3
        Subquery predicate ID:        Not Applicable
        Column Names:
        -----
        +Q5.TOTAL_AMOUNT+Q5.CUSTOMERNAME+Q5.CUSTOMERID

3) MSJOIN: (Merge Scan Join)
    Cumulative Total Cost:      13.6223
    Cumulative CPU Cost:       156871
    Cumulative I/O Cost:        2
    Cumulative Re-Total Cost:   13.6223
```

c) Merge Scan Join

```
Administrator:DB2 CLP - DB2COPY1
Column Names:
-----
+Q5.TOTAL_AMOUNT+Q5.CUSTOMERNAME+Q5.CUSTOMERID

3) MSJOIN: (Merge Scan Join)
   Cumulative Total Cost:      13.6223
   Cumulative CPU Cost:       106871
   Cumulative I/O Cost:        2
   Cumulative Re-total Cost:  13.6223
   Cumulative CPU Cost:       106871
   Cumulative Re-I/O Cost:     2
   Cumulative First Row Cost: 13.6218
   Estimated Bufferpool Buffers: 0

   Arguments:
   -----
   EARLQUIT: (Early Out flag)
      NONE
   INNERCOL: (Inner Order columns)
      1: Q1.CUSTOMERID(A)
   OUTERCOL: (Outer Order columns)
      1: Q2.CUSTOMERID(A)
   TEMPsize: (Temporary Table Page Size)
      8192

   Predicates:
   -----
   4) Predicate used in Join,
      Comparison Operator:      Equal (=)
      Subquery Input Required:  No
      Filter Factor:           0.166667

      Predicate Text:
      -----
      (Q2.CUSTOMERID = Q1.CUSTOMERID)

   Input Streams:
   -----
   4) From Operator #4
      Estimated number of rows:    6
      Number of columns:          2
      Subquery predicate ID:      Not Applicable
      Column Names:
      -----
      +Q2.CUSTOMERID(A)+Q2.CUSTOMERNAME(A)

   9) From Operator #7
      Estimated number of rows:    2
      Number of columns:          2
      Subquery predicate ID:      Not Applicable
      Column Names:
      -----
      +Q1.CUSTOMERID(A)+Q1.AMOUNT

   Output Streams:
   -----
```

d) Table Scan

```
Administrator:DB2 CLP - DB2COPY1
Column Names:
-----
+Q1.CUSTOMERID(A)+Q1.AMOUNT

Output Streams:
-----
10) To Operator #2
      Estimated number of rows:    12
      Number of columns:          3
      Subquery predicate ID:      Not Applicable
      Column Names:
      -----
      +Q3.CUSTOMERID(A)+Q3.CUSTOMERNAME(A)+Q3.AMOUNT

4) TBSCAN: (Table Scan)
   Cumulative Total Cost:      6.81836
   Cumulative CPU Cost:       68128
   Cumulative I/O Cost:        1
   Cumulative Re-total Cost:  6.800118802
   Cumulative CPU Cost:       68128
   Cumulative Re-I/O Cost:     0
   Cumulative First Row Cost: 6.81836
   Estimated Bufferpool Buffers: 0

   Arguments:
   -----
   ON INPUT: (Join input leg)
      OUTER
   MAXROWS: (Maximum pages for prefetch)
      NONE
   SCANDIR : (Scan Direction)
      FORWARD
   SPEED   : (Assumed speed of scan, in sharing structures)
      SLOW
   THROTTLE: (Scan may be throttled, for scan sharing)
   VISIBLE : (May be included in scan sharing structures)
      FALSE
   WRAPPING: (Scan may start anywhere and wrap)
      FALSE

   Input Streams:
   -----
   3) From Operator #5
      Estimated number of rows:    6
      Number of columns:          2
      Subquery predicate ID:      Not Applicable
      Column Names:
      -----
      +Q2.CUSTOMERID(A)+Q2.CUSTOMERNAME(A)

   Output Streams:
   -----
```

e) Sort

```
Administrator:DB2 CLP - DB2COPY1
Column Names:
+Q2.CUSTOMERID(A)+Q2.CUSTOMERNAME(A)

Output Streams:
-----)
4) To Operator #3
    Estimated number of rows:      6
    Number of columns:            2
    Subquery predicate ID:        Not Applicable

    Column Names:
    -----
    +Q2.CUSTOMERID(A)+Q2.CUSTOMERNAME(A)

5) SORT : (Sort)
    Cumulative Total Cost:      6.81026
    Cumulative CPU Cost:        600941
    Cumulative I/O Cost:         1
    Cumulative Re-total Cost:   0.00094894
    Cumulative Re-CPU Cost:    12054
    Cumulative Re-I/O Cost:     0
    Cumulative First Row Cost:  6.81026
    Estimated Bufferpool Buffers: 1

    Arguments:
    -----
    DUPLICATES : (Duplicates Warning flag)
        TRUE
    KEYS : (Key cardinality)
        6
    NUMROWS : (Estimated number of rows)
        6
    ROWWIDTH: (Estimated width of rows)
        20
    SORTKEYS : (Sort Key column)
        1: Q2.CUSTOMERID(A)
        2: Q2.CUSTOMERNAME(A)
    TEMPFILESIZE : (Temporary Table Page Size)
        8192
    UNIQUE : (Uniqueness required flag)
        FALSE

    Input Streams:
    -----
    2) From Operator #6
        Estimated number of rows:      6
        Number of columns:            2
        Subquery predicate ID:        Not Applicable

        Column Names:
        -----
        +Q2.CUSTOMERNAME+Q2.CUSTOMERID

    Output Streams:
    -----
    3) To Operator #4
```

```
Administrator:DB2 CLP - DB2COPY1
+Q2.CUSTOMERNAME+Q2.CUSTOMERID

Output Streams:
-----)
3) To Operator #4
    Estimated number of rows:      6
    Number of columns:            2
    Subquery predicate ID:        Not Applicable

    Column Names:
    -----
    +Q2.CUSTOMERID(A)+Q2.CUSTOMERNAME(A)

6) TBSCAN: (Table Scan)
    Cumulative Total Cost:      6.80993
    Cumulative CPU Cost:        62647
    Cumulative I/O Cost:         1
    Cumulative Re-total Cost:   0.00094894
    Cumulative Re-CPU Cost:    12054
    Cumulative Re-I/O Cost:     0
    Cumulative First Row Cost:  6.80993
    Estimated Bufferpool Buffers: 1

    Arguments:
    -----
    CUR_COMM: (Currently Committed)
    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_INDX : (Skip Inserted Rows)
        TRUE
    SPEED : (Assumed speed of scan, in sharing structures)
        FAST
    TABLOCK : (Table Lock Intent)
        INTENT SHARE
    TBISOLVL: (Table access Isolation level)
        READ 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 XINFERNAL.CUSTOMER
        Estimated number of rows:      6
        Number of columns:            3
        Subquery predicate ID:        Not Applicable

        Column Names:
```

f) Filter

```

Administrator:DB2 CLP - DB2COPY1
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 XINFERNAL.CUSTOMER
    Estimated number of rows:      6
    Number of columns:            3
    Subquery predicate ID:        Not Applicable
    Column Names:
    -----
    +Q2.SRID$+Q2.CUSTOMERNAME+Q2.CUSTOMERID

Output Streams:
-----
2) To Operator #5
    Estimated number of rows:      6
    Number of columns:            2
    Subquery predicate ID:        Not Applicable
    Column Names:
    -----
    +Q2.CUSTOMERNAME+Q2.CUSTOMERID

7) FILTER: (Filter)
    Cumulative Total Cost:      6.81128
    Cumulative CPU Cost:        79772.7
    Cumulative I/O Cost:         1
    Cumulative Re-total Cost:   0.00211949
    Cumulative Re-CPU Cost:    26923
    Cumulative Re-I/O Cost:     0
    Cumulative First Row Cost: 6.81128
    Estimated Bufferpool Buffers: 0
    Arguments:
    -----
    IN INPUT: (Join input leg)
    INNER
    Predicates:
    -----
    4) Residual Predicate,
        Comparison Operator:      Equal (=)
        Subquery Input Required:  No
        Filter Factor:           0.166667
        Predicate Text:
        -----
        (Q2.CUSTOMERID = Q1.CUSTOMERID)

Input Streams:
-----
8) From Operator #8
Administrator:DB2 CLP - DB2COPY1... 272_Ravin.docx - Word
Windows Taskbar: edge computing secu... (no subject) - rovin28... Steam Login MISC
Administrator:DB2 CLP - DB2COPY1... 272_Ravin.docx - Word
8:48 PM
-----
```

```

Administrator:DB2 CLP - DB2COPY1
(Q2.CUSTOMERID = Q1.CUSTOMERID)

Input Streams:
-----
8) From Operator #8
    Estimated number of rows:      12
    Number of columns:            2
    Subquery predicate ID:        Not Applicable
    Column Names:
    -----
    +Q1.CUSTOMERID(A)+Q1.AMOUNT

Output Streams:
-----
9) To Operator #3
    Estimated number of rows:      2
    Number of columns:            2
    Subquery predicate ID:        Not Applicable
    Column Names:
    -----
    +Q1.CUSTOMERID(A)+Q1.AMOUNT

8) TBSHOW: (Table Scan)
    Cumulative Total Cost:      6.81128
    Cumulative CPU Cost:        79772.7
    Cumulative I/O Cost:         1
    Cumulative Re-total Cost:   0.00211949
    Cumulative Re-CPU Cost:    26923
    Cumulative Re-I/O Cost:     0
    Cumulative First Row Cost: 6.81128
    Estimated Bufferpool Buffers: 0
    Arguments:
    -----
    MAXPAGES: (Maximum pages for prefetch)
    PREFETCH: (Type of Prefetch)
        NONE
    SCANDIR : (Scan Direction)
        FORWARD
    SPEED : (Assumed speed of scan, in sharing structures)
        SLOW
    THROTTLE: (Scan may be throttled, for scan sharing)
        FALSE
    VISIBLE: (May be included in scan sharing structures)
        FALSE
    WRAPPING: (Scan may start anywhere and wrap)
        FALSE

Input Streams:
-----
7) From Operator #9
    Estimated number of rows:      12
    Number of columns:            2
    Subquery predicate ID:        Not Applicable
Administrator:DB2 CLP - DB2COPY1... 272_Ravin.docx - Word
Windows Taskbar: edge computing secu... (no subject) - rovin28... Steam Login MISC
Administrator:DB2 CLP - DB2COPY1... 272_Ravin.docx - Word
8:51 PM
-----
```

```

Administrator:DB2 CLP - DB2COPY1
WRAPPING: (Scan may start anywhere and wrap)
    FALSE

Input Streams:
----- 7) From Operator #9
    Estimated number of rows:      12
    Number of columns:           2
    Subquery predicate ID:       Not Applicable
    Column Names:
    -----
    +Q1.CUSTOMERID(A)+Q1.AMOUNT

Output Streams:
----- 8) To Operator #7
    Estimated number of rows:      12
    Number of columns:           2
    Subquery predicate ID:       Not Applicable
    Column Names:
    -----
    +Q1.CUSTOMERID(A)+Q1.AMOUNT

9) SORT : (Sort)
    Cumulative Total Cost:      6.81118
    Cumulative CPU Cost:        70465.7
    Cumulative I/O Cost:          1
    Cumulative Re-Total Cost:   0.00176232
    Cumulative Re-CPU Cost:    22356
    Cumulative Re-I/O Cost:     0
    Cumulative First Row Cost:  6.81118
    Estimated Bufferpool Buffers: 1

    Arguments:
    -----
    DUPLWARN: (Duplicates Warning Flag)
        FALSE
    KEYS : (Key cardinality)
        5
    NUMROWS : (Estimated number of rows)
        12
    ROWWIDTH: (Estimated width of rows)
        16,000
    SORTKEY: (Sort Key column)
        1: Q1.CUSTOMERID(A)
    TEMPSIZE: (Temporary Table Page Size)
        8192
    UNIQUE : (Uniqueness required flag)
        FALSE

Input Streams:
----- 6) From Operator #10
    Estimated number of rows:      12
    Number of columns:           2
    Subquery predicate ID:       Not Applicable

Administrator:DB2 CLP - DB2COPY1
FALSE

Input Streams:
----- 6) From Operator #10
    Estimated number of rows:      12
    Number of columns:           2
    Subquery predicate ID:       Not Applicable
    Column Names:
    -----
    +Q1.AMOUNT+Q1.CUSTOMERID

Output Streams:
----- 7) To Operator #8
    Estimated number of rows:      12
    Number of columns:           2
    Subquery predicate ID:       Not Applicable
    Column Names:
    -----
    +Q1.CUSTOMERID(A)+Q1.AMOUNT

10) TBSCAN: (Table Scan)
    Cumulative Total Cost:      6.81074
    Cumulative CPU Cost:        72979
    Cumulative I/O Cost:          1
    Cumulative Re-Total Cost:   0.00162532
    Cumulative Re-CPU Cost:    22386
    Cumulative Re-I/O Cost:     0
    Cumulative First Row Cost:  6.80912
    Estimated Bufferpool Buffers: 1

    Arguments:
    -----
    CUR_COMMIT: (Currently Committed)
        TRUE
    LOCKVOID: (Lock Avoidance)
    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
    TABLELOCK : (Table Lock intent)
        INTERNAL
    TBSOLVE: (Table Access Isolation Level)
        CURSOR STABILITY
    THROTTLE: (Scan may be throttled, for scan sharing)
        TRUE
    VISIBLE : (May be included in scan sharing structures)
        TRUE

```

- Objects used in Access Plan

```
Administrator:DB2 CLP - DB2COPY1
SPEED : (Assumed speed of scan, In sharing structures)
FAST
TABLELOCK : (Table lock intent)
INTENT SHARE
TBSOLVL: (Table access Isolation Level)
CURSOR STABILITY
THROTTLED: (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:
-----
5) From Object XINFERMAL_PAYMENT
   Estimated number of rows:      12
   Number of columns:            3
   Subquery predicate ID:        Not Applicable
   Column Names:
   -----
   +Q1.RID$+Q1.AMOUNT+Q1.CUSTOMERID

Output Streams:
-----
6) To Operator #9
   Estimated number of rows:      12
   Number of columns:            3
   Subquery predicate ID:        Not Applicable
   Column Names:
   -----
   +Q1.AMOUNT+Q1.CUSTOMERID

Objects Used in Access Plan:
-----
Schema: XINFERMAL
Name: CUSTOMER
Type: Table
Time of creation: 2017-02-21-18:37:00.188001
Last statistics update: 2017-02-21-20:08:25.666000
Number of columns: 2
Number of rows: 6
Width of rows: 22
Number of buffer pool pages: 1
Number of data partitions: 1
Distinct row values: No
Tablespace name: TB0025AMPLEREL
Tablespace overhead: 6.725000
Tablespace transfer rate: 0.080000
Source for statistics: Single Node
Prefetch page count: 32
Container extent page count: 32
Table overflow record count: 0
Table Active Blocks: 1
Average Row Compression Ratio: 0
Percentage Rows Compressed: 0
Average Compressed Row Size: 0

Administrator:DB2 CLP - DB2COPY1... 272_Ravin.docx - Word
```

```
Administrator:DB2 CLP - DB2COPY1
Objects Used in Access Plan:
-----
Schema: XINFERMAL
Name: CUSTOMER
Type: Table
Time of creation: 2017-02-21-18:37:00.188001
Last statistics update: 2017-02-21-20:08:25.666000
Number of columns: 2
Number of rows: 6
Width of rows: 22
Number of buffer pool pages: 1
Number of data partitions: 1
Distinct row values: No
Tablespace name: TB0025AMPLEREL
Tablespace overhead: 6.725000
Tablespace transfer rate: 0.080000
Source for statistics: Single Node
Prefetch page count: 32
Container extent page count: 32
Table overflow record count: 0
Table Active Blocks: 1
Average Row Compression Ratio: 0
Percentage Rows Compressed: 0
Average Compressed Row Size: 0

Schema: XINFERMAL
Name: PAYMENT
Type: Table
Time of creation: 2017-02-21-18:42:00.004001
Last statistics update: 2017-02-21-20:08:25.078000
Number of columns: 4
Number of rows: 12
Width of rows: 42
Number of buffer pool pages: 1
Number of data partitions: 1
Distinct row values: No
Tablespace name: TB0025AMPLEREL
Tablespace overhead: 6.725000
Tablespace transfer rate: 0.080000
Source for statistics: Single Node
Prefetch page count: 32
Container extent page count: 32
Table overflow record count: 0
Table Active Blocks: 1
Average Row Compression Ratio: 0
Percentage Rows Compressed: 0
Average Compressed Row Size: 0

Executing Connect Reset -- Connect Reset was Successful.
C:\Program Files\IBM\SQLLIB\MISC\
```

3. Graph Data Store Assignment

Creating a Graph

Step 1: Generating Session Token used for Authentication in subsequent Request.

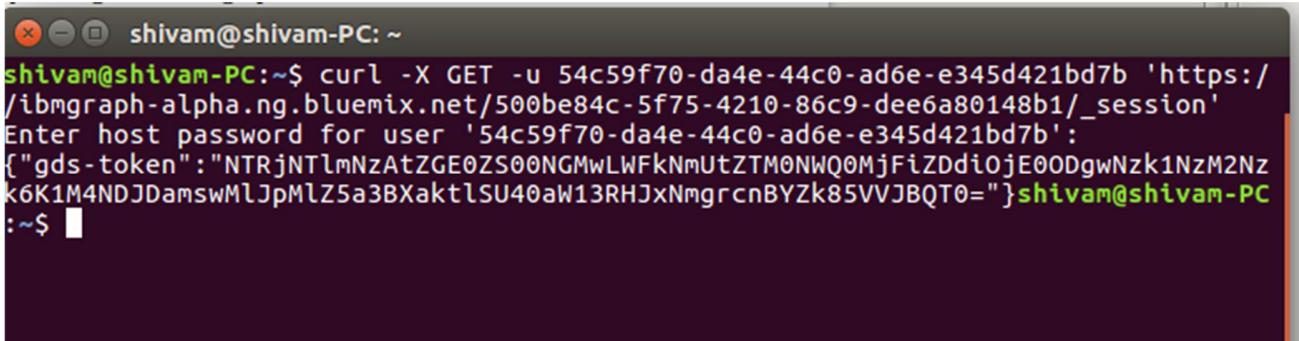
Method	URI	Request	Response	Description
GET	/\$BASE_URL/_session		{ "gds-token": "OTNiYjk..."} [REDACTED]	Returns a session token that you can pass into subsequent requests.

Use Session API along with the user-name to get Token.

Eg :

```
shivam@shivam-PC:~$ curl -X GET -u 54c59f70-da4e-44c0-ad6e-e345d421bd7b 'https://ibmgraph-alpha.ng.bluemix.net/500be84c-5f75-4210-86c9-dee6a80148b1/_session'
```

Then Enter the Password to receive the Token.



```
shivam@shivam-PC: ~
shivam@shivam-PC:~$ curl -X GET -u 54c59f70-da4e-44c0-ad6e-e345d421bd7b 'https://ibmgraph-alpha.ng.bluemix.net/500be84c-5f75-4210-86c9-dee6a80148b1/_session'
Enter host password for user '54c59f70-da4e-44c0-ad6e-e345d421bd7b':
{"gds-token": "NTRjNTlmNzAtZGE0ZS00NGMwLWFkNmUtZTM0NWQ0MjFiZDdiOjE0ODgwNzk1NzM2Nz
k6K1M4NDJDamswMlJpMlZ5a3BXaktlSU40aW13RHJxNmgrcnBYZk85VVJBQT0="}shivam@shivam-PC
:~$
```

Step 2: Creating an Alias for Curl + GDS Token, for simplicity.

```
shivam@shivam-PC:~$ alias acurl='curl -H "Authorization : gds-token :
NTRjNTlmNzAtZGE0ZS00NGMwLWFkNmUtZTM0NWQ0MjFiZDdiOjE0ODgwNzk1NzM2Nz
k6K1M4NDJDamswMlJpMlZ5a3BXaktlSU40aW13RHJxNmgrcnBYZk85VVJBQT0=""'
```

```
shivam@shivam-PC:~$ alias acurl='curl -H "Authorization : gds-token : NTRjNTlmNzAtZGE0ZS00NGMwLWFkNmUtZTM0NWQ0MjFzDdi0jE0ODgwNzk1NzM2Nzk6K1M4NDJDamswMlJpMlZ5a3BXaktlsU40aW13RHJxNmgrcnBYZk85VVJBQT0="'
shivam@shivam-PC:~$
```

Step 3: Creating a New Graph.

```
curl "https://ibmgraph-alpha.ng.bluemix.net/500be84c-5f75-4210-86c9-dee6a80148b1/_graphs/flywirelikedemo"\n-X POST \
-H "Authorization: gds-token"
OTdLYjFhMDItMzZkNS00MWJmLWEyNTItNjMxYzI3Zdk5ZGZmOjE0ODgwODAyMTY5NjE6Z1pnRTdzTE1
DMzUyMHVaMmc3SzUvanJtVzMzTkdOd3BMaE9aRUV4Z09IST0="
```

```
shivam@shivam-PC:~$ curl "https://ibmgraph-alpha.ng.bluemix.net/500be84c-5f75-4210-86c9-dee6a80148b1/_graphs/flywirelikedemo"\n> -X POST \
> -H "Authorization: gds-token 0TdLYjFhMDItMzZkNS00MWJmLWEyNTItNjMxYzI3Zdk5ZGZmOjE0ODgwODAyMTY5NjE6Z1pnRTdzTE1DMzUyMHVaMmc3SzUvanJtVzMzTkdOd3BMaE9aRUV4Z09IST0="
{"graphId":"flywirelikedemo","dbUrl":"https://ibmgraph-alpha.ng.bluemix.net/500be84c-5f75-4210-86c9-dee6a80148b1/flywirelikedemo"}shivam@shivam-PC:~$
```

Step 4: Creating Schema:

- To Create an Schema run the following Command in the terminal:

```
curl -X POST \
"https://ibmgraph-alpha.ng.bluemix.net/500be84c-5f75-4210-86c9-dee6a80148b1/flywirelikedemo/schema" \
-H "Content-Type:application/json" \
-H "authorization:gds-token"
OTdLYjFhMDItMzZkNS00MWJmLWEyNTItNjMxYzI3Zdk5ZGZmOjE0ODgwODAyMTY5NjE6Z1pnRTdzTE1
DMzUyMHVaMmc3SzUvanJtVzMzTkdOd3BMaE9aRUV4Z09IST0=" \
-d '{
  "propertyKeys": [
    {"name": "Name", "dataType": "String", "cardinality": "SINGLE"},  

    {"name": "Amount_recieved", "dataType": "Integer", "cardinality": "SINGLE"},  

    {"name": "Status", "dataType": "Boolean", "cardinality": "SINGLE"}]
```

```

{
  "name": "University_ID", "dataType": "Integer", "cardinality": "SINGLE"},  

  {"name": "Student_id", "dataType": "String", "cardinality": "SINGLE"},  

  {"name": "University_name", "dataType": "String", "cardinality": "SINGLE"},  

  {"name": "Customer_Name", "dataType": "String", "cardinality": "SINGLE"},  

  {"name": "Ammount_to_be_delivered", "dataType": "Integer", "cardinality": "SINGLE"},  

  {"name": "ID", "dataType": "Integer", "cardinality": "SINGLE"},  

  {"name": "Amount_rec", "dataType": "String", "cardinality": "SINGLE"},  

  {"name": "Third_party_name", "dataType": "String", "cardinality": "SINGLE"},  

  {"name": "Time", "dataType": "String", "cardinality": "SINGLE"}  

],  

  "vertexLabels": [  

    {"name": "Student"},  

    {"name": "Flywire"},  

    {"name": "University"}  

],  

  "edgeLabels": [  

    {"name": "pay_broker", "multiplicity": "ONE2ONE"},  

    {"name": "pay_university", "multiplicity": "ONE2ONE"}  

],  

  "vertexIndexes": [  

    {"name": "vByName", "propertyKeys": ["Name"], "composite": true, "unique": true},  

    {"name": "vByAmount_recieved", "propertyKeys": ["Amount_recieved"], "composite": true, "unique": false},  

    {"name": "vByUniversity_ID", "propertyKeys": ["University_ID"], "composite": true, "unique": false},  

    {"name": "vByStudent_id", "propertyKeys": ["Student_id"], "composite": true, "unique": true},  

    {"name": "vByUniversity_name", "propertyKeys": ["University_name"], "composite": true, "unique": false},  

    {"name": "vByCustomer_Name", "propertyKeys": ["Customer_Name"], "composite": true, "unique": true},  

    {"name": "vByAmmount_to_be_delivered", "propertyKeys": ["Ammount_to_be_delivered"], "composite": true, "unique": false},  

    {"name": "vByID", "propertyKeys": ["ID"], "composite": true, "unique": true},  

    {"name": "vByAmount_rec", "propertyKeys": ["Amount_rec"], "composite": true, "unique": false},  

    {"name": "vByThird_party_name", "propertyKeys": ["Third_party_name"], "composite": true, "unique": false}  

],  

  "edgeIndexes": [  

    {"name": "eByTime", "propertyKeys": ["Time"], "composite": true, "unique": false}  

  ] }  


```

Below is the Output of the Above Script Execution.

```
{"requestId":"8c7242b2-aeb8-4668-aafe-7f8d7372b5fe","status":{"message":"","code":200,"attributes":{}}, "result":{"data":[{"propertyKeys":[{"name":"Name","dataType":"String","cardinality":"SINGLE"}, {"name":"Amount_recieved","dataType":"Integer","cardinality":"SINGLE"}, {"name":"University_ID","dataType":"Integer","cardinality":"SINGLE"}, {"name":"Student_id","dataType":"String","cardinality":"SINGLE"}, {"name":"University_name","dataType":"String","cardinality":"SINGLE"}, {"name":"Customer_Name","dataType":"String","cardinality":"SINGLE"}, {"name":"Ammount_to_be_delivered","dataType":"Integer","cardinality":"SINGLE"}, {"name":"ID","dataType":"Integer","cardinality":"SINGLE"}, {"name":"Amount_rec","dataType":"String","cardinality":"SINGLE"}, {"name":"Third_party_name","dataType":"String","cardinality":"SINGLE"}, {"name":"Time","dataType":"String","cardinality":"SINGLE"}], "vertexLabels":[{"name":"Student"}, {"name":"Flywire"}, {"name":"University"}], "edgeLabels":[{"name":"payBroker","directed":true,"multiplicity":"ONE2ONE"}, {"name":"payUniversity","directed":true,"multiplicity":"ONE2ONE"}], "vertexIndexes":[{"name":"vByName","composite":true,"unique":true,"propertyKeys":["Name"], "requiresReindex":false,"type":"vertex"}, {"name":"vByAmount_recieved","composite":true,"unique":false,"propertyKeys":["Amount_recieved"], "requiresReindex":false,"type":"vertex"}, {"name":"vByUniversity_ID","composite":true,"unique":false,"propertyKeys":["University_ID"], "requiresReindex":false,"type":"vertex"}, {"name":"vByStudent_id","composite":true,"unique":true,"propertyKeys":["Student_id"], "requiresReindex":false,"type":"vertex"}, {"name":"vByUniversity_name","composite":true,"unique":false,"propertyKeys":["University_name"], "requiresReindex":false,"type":"vertex"}, {"name":"vByCustomer_Name","composite":true,"unique":true,"propertyKeys":["Customer_Name"], "requiresReindex":false,"type":"vertex"}, {"name":"vByAmmount_to_be_delivered","composite":true,"unique":false,"propertyKeys":["Ammount_to_be_delivered"], "requiresReindex":false,"type":"vertex"}, {"name":"vByID","composite":true,"unique":true,"propertyKeys":["ID"], "requiresReindex":false,"type":"vertex"}, {"name":"vByAmount_rec","composite":true,"unique":false,"propertyKeys":["Amount_rec"], "requiresReindex":false,"type":"vertex"}, {"name":"vByThird_party_name","composite":true,"unique":false,"propertyKeys":["Third_party_name"], "requiresReindex":false,"type":"vertex"}], "edgeIndexes":[{"name":"eByTime","composite":true,"unique":false,"propertyKeys":["Time"], "requiresReindex":false,"type":"edge"}]}], "meta":{}}}}$
```

Above Response shows that the Schema was successfully created and gives details of the Graph Element Created.

Now Since our Schema is created now we are ready to feed our Data base with data. Following are the steps to Load Data into the Graph.

Step 5: Filling Up the Data in the data Points.

Creating an Json file for small Input.

```

shivam@shivam-PC:~$ cat << ENDGERMLIN > gremlin.json
> [
>   "gremlin":"
>   def shivam = graph.addVertex(t.label,'Student','Name','Shivam Kumar Gupta','Amount_recieved','7500','University_ID','01','Student_id',011545196);
>   def Shivam_to_flywire = graph.addVertex(t.label,'Flywire','Customer_Name','Shivam Kumar Gupta','University_name','SJSU','Ammount_to_be_delivered',7450,);
>   def flywire_to_univ = graph.addVertex(t.label,'University','ID',011545196,'Amount_rec','7450','Third_party_name','flywire');
>   shivan.addEdge('payBroker','Time','02_01_2017');
>   "
>   }
>   "
> ENDGERMLIN
shivam@shivam-PC:~$ █

```

For Bulk entry we can follow following URL.

<https://ibm-graph-docs.ng.bluemix.net/api.html#bulk-input-apis>

Since we have created JSON file now we can go ahead and load the data into the Graph using following script:

```

curl -X POST \
"https://ibmgraph-alpha.ng.bluemix.net/500be84c-5f75-4210-86c9-
dee6a80148b1/flywirelikedemo/gremlin" \
-H "Content-Type:application/json" \
-H "authorization:gds-token █
OTdIYjFhMDItMzKNS00MWJmLWEyNTItNjMxYzI3ZDk5ZGZmOjE0ODgwODAyMTY5NjE6Z1
pnRTdzTE1DMzUyMHVaMmc3SzUvanJtVzMzTkdOd3BMaE9aRUV4Z09lST0=" \
-d @gremlin.json █

```

And the response of the above script shows that our Data was loaded properly in the Graph.

```

shivam@shivam-PC:~$ curl -X POST \
>   "https://ibmgraph-alpha.ng.bluemix.net/500be84c-5f75-4210-86c9-dee6a80148b1/flywirelikedemo/gremlin" \
>   -H "Content-Type:application/json" \
>   -H "authorization:gds-token OTdIYjFhMDItMzKNS00MWJmLWEyNTItNjMxYzI3ZDk5ZGZmOjE0ODgwODAyMTY5NjE6Z1pnRTdzTE1DMzUyMHVaMmc3SzUvanJtVzMzTkdOd3BMa
E9aRUV4Z09lST0=" \
>   -d @gremlin.json
{"requestId":"abad0364-892c-4eea-a3fe-d938303b7b14","status":{"message":"","code":200,"attributes":{}}, "result":{"data":[{"id":4112,"label":"University","type":"vertex","properties":{"Amount_rec":[{"id":"1ky-368-pad","value":"7450"}],"Third_party_name":[{"id":"1z6-368-q2t","value":"flywire"}]},{ "ID": [{"id": "16q-368-ohx", "value": "9"}]}]}, "meta":{}} shivam@shivam-PC:~$ █

```