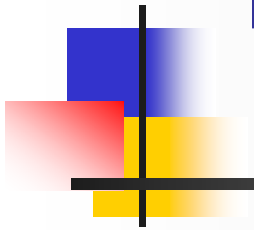
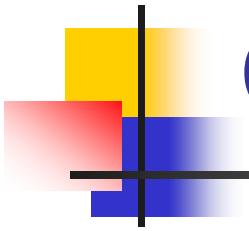


EasyQL

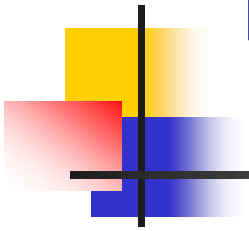
By: Kangkook Jee
Saahil Peerbhoy
Smridh Thapar
Kishan Iyer



Our Goals:

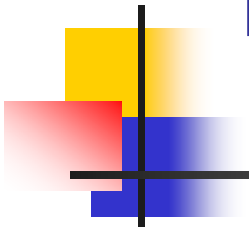
- Database Manipulation Language
- May connect to multiple databases
- Single environment from which to manage inf ormation from different databases

EasyQL Overview



- Allows common SQL operations
- Procedural programming and control flow
- Syntax “inspired” by Java and SQL

EasyQL Syntax and Semantics



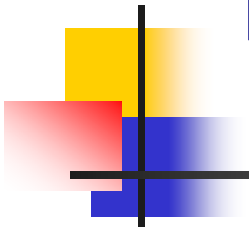
- Data types: int, float, varchar, table, and connect ion
- If-else and while are supported
- “[ ]” for attributes, “( )” for conditions

EasyQL Connection

connection c1;

c1 = conn("<host_add>", "<port>", "<dbna
me>", "<dbtype>", "<username>",
"<password>");

EasyQL Table setup



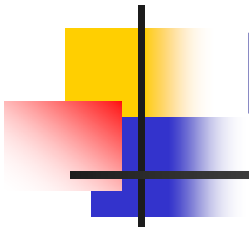
- Table declaration: table type variables initialized to “garbage” value.
- Create table:
  - Tab1.create(<conn_id>, “<attributes>”);
- Table assignment:
  - tab1 = C1::coffee ;
  - tab2 = tab1[“<attribute_names>”](“<conditions>”);

Table operations:

- Update:
  - tab1.update[“<assignments>”](“<conditions>”);
- Insert:
  - Tab1.insert[“<attributes>”](“<values>”);
- Delete:
  - Tab1.delete(“<conditions>”);

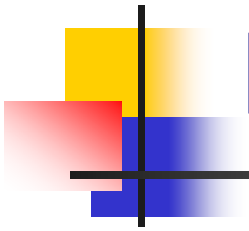
Example code (if, while)

```
int i,j;

i=0;
j=10;

while (j>i)
{
        display ("i is");
        display (i);
        display ("j is");
        display (j);

        i=i+1;

        if (i==10) {
                display ("i is 10, exiting");
        }
}
```



Example code (create table)

```
table T1;
int ret;

ret = T1.create (DefConn,"name varchar(255), age int");

if (ret)  {
        T1=DefConn::T1;

        display ("");
        display ("Tables in Default Connection");
        display (DefConn.list());

        display ("");
        display ("Description of T1");
        display (T1);
} else {
        display ("create table failed");
}
```



Example code (table desc)

```
connection c1;
int i1;
table t1;

c1 = conn("69.22.220.234", "3306", "test", "mysql", "root", "plt123");

t1 = c1::COFFEES;

display ("");
display ("### Table metadata printing ###");
display ("");
display (t1.dsc());

display ("");
display ("### Table selecting ###");
display (t1["COF_NAME,SUP_ID"]("SALES=99"));
```

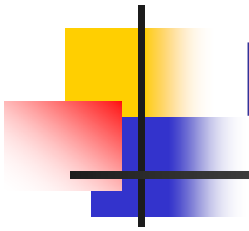
Example code (insert, delete, update)

```
connection c1;
int i1;
table t1,t2;


c1 = conn("69.22.220.234", "3306", "test", "mysql", "root", "plt123");
i1=0;
t1.create (c1,"name varchar(255), age int");
t2=c1::t1;

display ("");
display ("#### Before insert ####");
display ("");

display (t2);
…
```

Example code (insert, delete, update) - Cont.

```
display ("");
display ("#### inserting  ####");
display ("");

while (i1<5) {
        i1=i1+1;
        display (i1);
        t2.insert["name,age"]("'Kishan',22");
}

i1=0;

while (i1<5) {
        i1=i1+1;
        display (i1);
        t2.insert["name,age"]("'Saahil',23");
}
```

```
display ("");
display ("#### After insert ####");
display ("");

display (t2);

t2.update ["name='kangkook'"]("age=22");

display ("");
display ("#### After update ####");
display ("");

display (t2);

t2.delete ("name='kangkook'");

display ("");
display ("#### After deletion ####");
display ("");

display (t2);

t2.store (c1);
```



Example code (store)

```
connection c1,c2;
int i1;
table t1;

c1 = conn("69.22.220.234", "3306", "test", "mysql", "root", "plt123");
c2 = conn("localhost", "3306", "test", "mysql", "root", "plt123");

display ("");
display ("table t1 from connection c1");
display ("");

t1=c1::t2;
display (t1);

display ("");
display ("table t1 from connection c2");
display ("");
display (c2::t1);

t1.store (c2);
display (c2::t1);
```

Division of Labour

- Kangkook: Front-end, back-end, testing
- Kishan: Front-end, documentation
- Smridh: Back-end, documentation, testing
- Saahil : Front-end, documentation, testing