University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 4

Date: June 22nd

Group Number: 7

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Emma Park (Heayoung)	33281130	y6f0b	emma95@student.ubc.ca
Shumin Wang	70072111	g5o9x	shumin11@student.ubc.ca
Mingyue (Miranda) Tang	13159264	g0v3o	mtang78@student.ubc.ca

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

University of British Columbia, Vancouver
Department of Computer Science

Deliverables

Commit the deliverables below to the CPSC 304 provided repository and submit a link to your repository on Canvas.

Repository Link:
https://github.students.cs.ubc.ca/CPSC304-2023S-T1/project_g0v3o_g5o9x_y6f0b.git

Link to webpage from demo:
https://www.students.cs.ubc.ca/~shumin11/labInventory/controller/main.php

1. **A short description of the final project, and what it accomplished.**

   Our project encompasses a comprehensive Lab Inventory Database designed to address the challenges researchers face in managing laboratory supplies and equipment. It focuses on tracking, organizing, and maintaining the stock of chemicals and equipment used in laboratory operations.

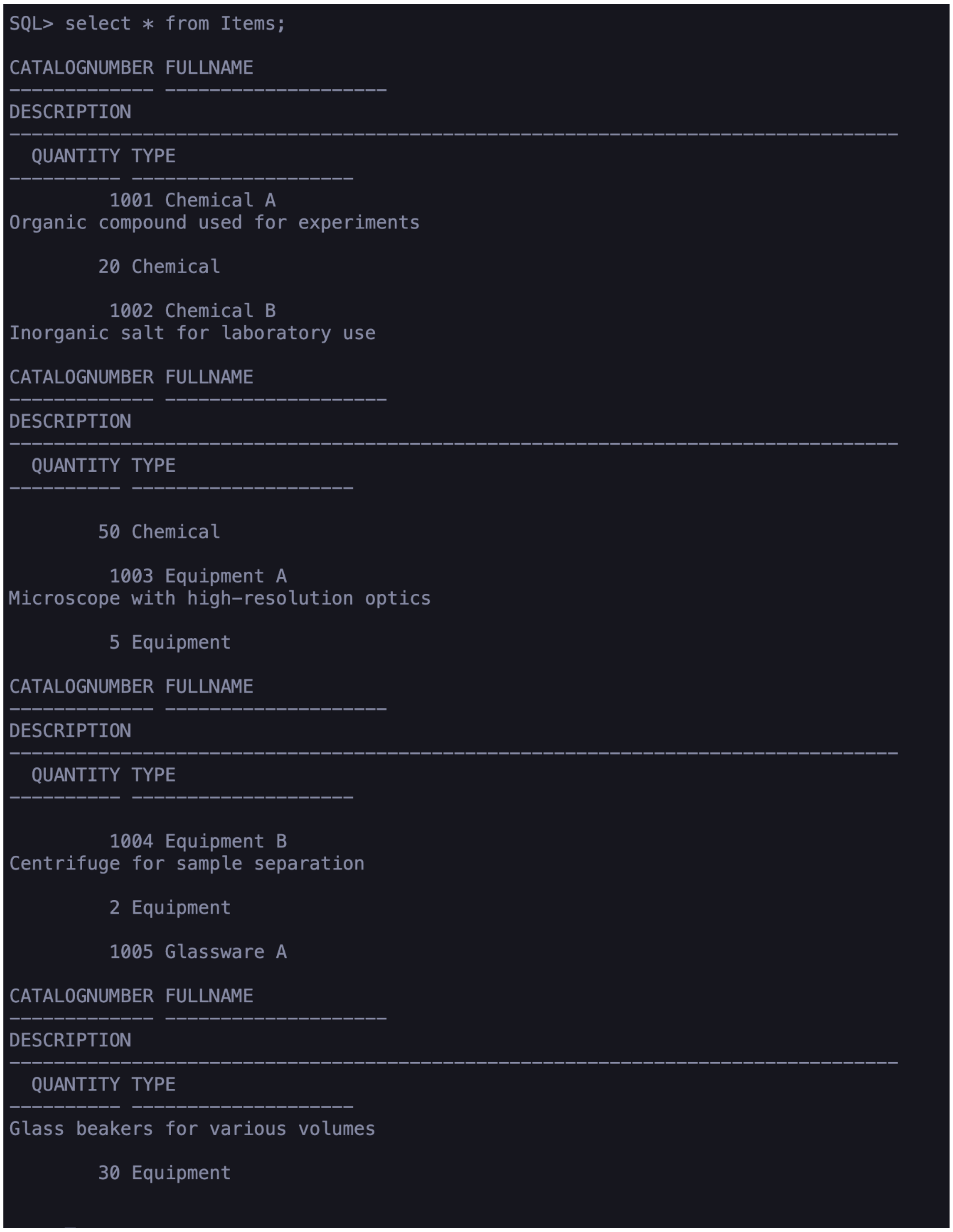
   Furthermore, this database-driven application efficiently manages and analyzes purchase, vendor, and current stock data. It seamlessly connects Purchase and Current Stock data, enabling effortless identification of items purchased from specific vendors. Users can evaluate vendor performance by counting purchases and identify those with significant purchase volumes. The application supports advanced analysis through nested aggregation, displaying items with quantities below the average. Additionally, it streamlines the tracking of purchase dates from all vendors.

2. **A description of how your final schema differed from the schema you turned in.**
   a. i. If the final schema differed, explain why. Note that turning in a final schema that's different from what you planned is fine, we just want to know what changed and why.

   To make the database more adaptable for future changes and relationships, we added a "Type" attribute with a data type of CHAR(20) to the Items Table. This allows for categorizing items into different types, such as Chemicals, Equipment, or future additions, making it easier to manage and analyze the inventory. The CHAR(20) data type ensures enough space to store descriptive type labels for items.

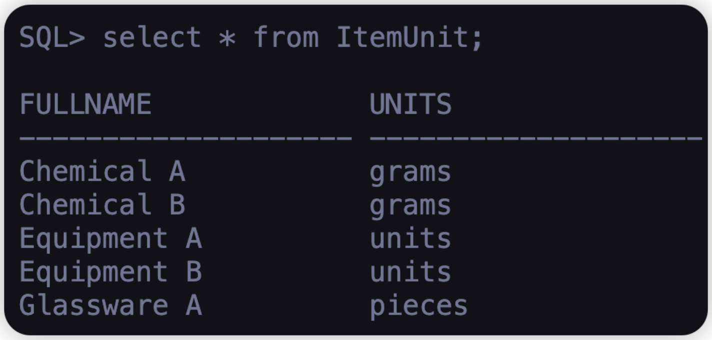
3. **A copy of the schema and screenshots that show what data is present in each relation after the SQL script from item #2 is run.**

- Items(<u>CatalogNumber</u>, FullName, Description, Quantity, Type)



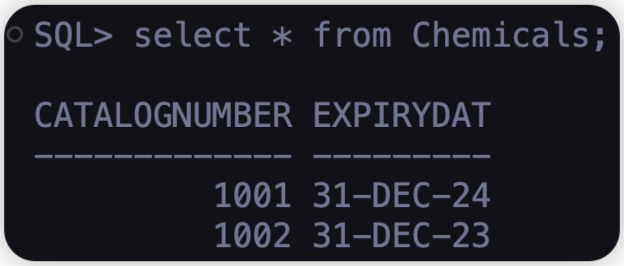
```
SQL> select * from Items;

CATALOGNUMBER FULLNAME
------------- --------------------
DESCRIPTION
--------------------------------------------------------------------------------
  QUANTITY TYPE
---------- --------------------
         1001 Chemical A
Organic compound used for experiments

        20 Chemical

         1002 Chemical B
Inorganic salt for laboratory use

CATALOGNUMBER FULLNAME
------------- --------------------
DESCRIPTION
--------------------------------------------------------------------------------
  QUANTITY TYPE
---------- --------------------

        50 Chemical

         1003 Equipment A
Microscope with high-resolution optics

         5 Equipment

CATALOGNUMBER FULLNAME
------------- --------------------
DESCRIPTION
--------------------------------------------------------------------------------
  QUANTITY TYPE
---------- --------------------

         1004 Equipment B
Centrifuge for sample separation

         2 Equipment

         1005 Glassware A

CATALOGNUMBER FULLNAME
------------- --------------------
DESCRIPTION
--------------------------------------------------------------------------------
  QUANTITY TYPE
---------- --------------------
Glass beakers for various volumes

        30 Equipment
```

- ItemUnit (<u>FullName</u>, Units)



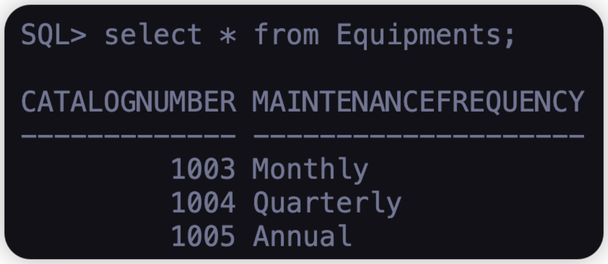
```
SQL> select * from ItemUnit;

FULLNAME             UNITS
-------------------- --------------------
Chemical A           grams
Chemical B           grams
Equipment A          units
Equipment B          units
Glassware A          pieces
```

- Chemicals (**<u>CatalogNumber</u>**, ExpiryDate)



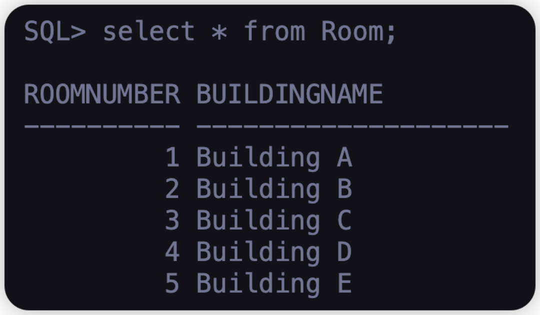
```
SQL> select * from Chemicals;

CATALOGNUMBER EXPIRYDAT
------------- ---------
         1001 31-DEC-24
         1002 31-DEC-23
```

- Equipments (**<u>CatalogNumber</u>**, MaintenanceFrequency)



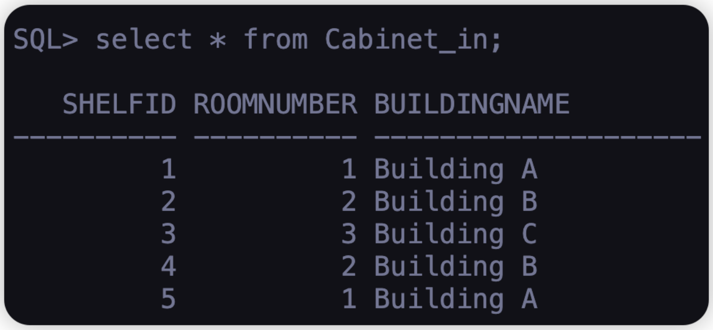
```
SQL> select * from Equipments;

CATALOGNUMBER MAINTENANCEFREQUENCY
------------- --------------------
         1003 Monthly
         1004 Quarterly
         1005 Annual
```

- Room (<u>RoomNumber</u>, <u>BuildingName</u>)



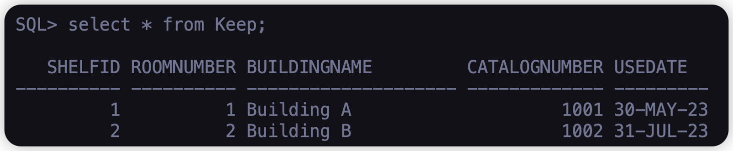
```
SQL> select * from Room;

ROOMNUMBER BUILDINGNAME
---------- --------------------
         1 Building A
         2 Building B
         3 Building C
         4 Building D
         5 Building E
```

- Cabinet_In (ShelfID, **RoomNumber**, **BuildingName**)



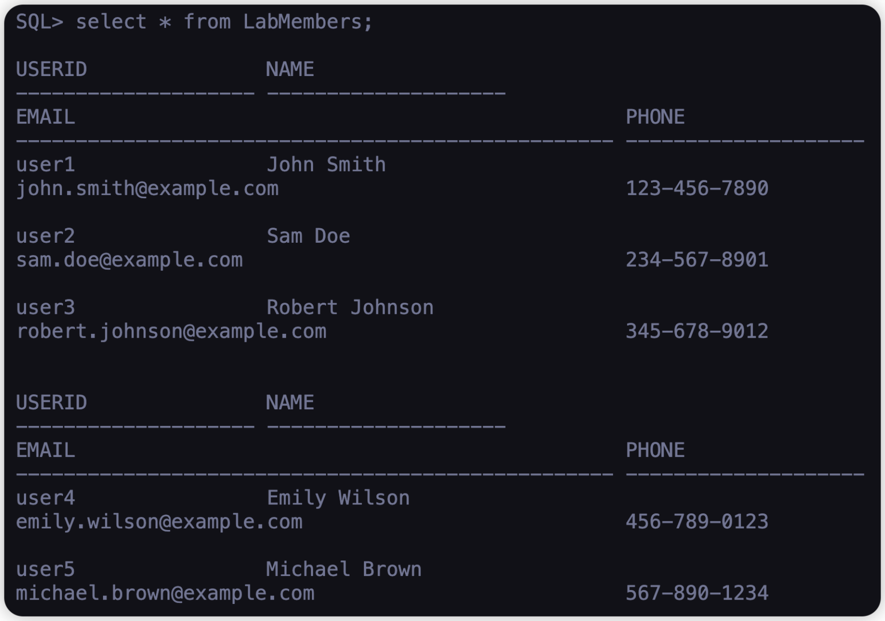
```
SQL> select * from Cabinet_in;

   SHELFID ROOMNUMBER BUILDINGNAME
---------- ---------- --------------------
         1          1 Building A
         2          2 Building B
         3          3 Building C
         4          2 Building B
         5          1 Building A
```

- Keep (**ShelfID**, **RoomNumber**, **BuildingName**, **CatalogNumber**, UseDate)



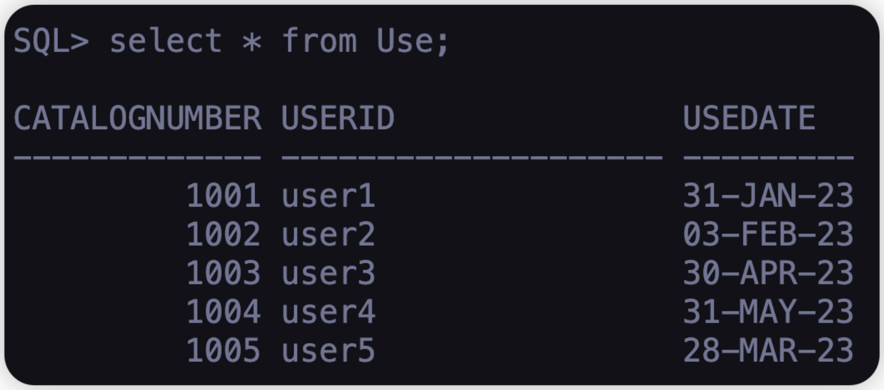
```
SQL> select * from Keep;

   SHELFID ROOMNUMBER BUILDINGNAME         CATALOGNUMBER USEDATE
---------- ---------- -------------------- ------------- ---------
         1          1 Building A                    1001 30-MAY-23
         2          2 Building B                    1002 31-JUL-23
```

- LabMembers (UserID , Name, Email, Phone)



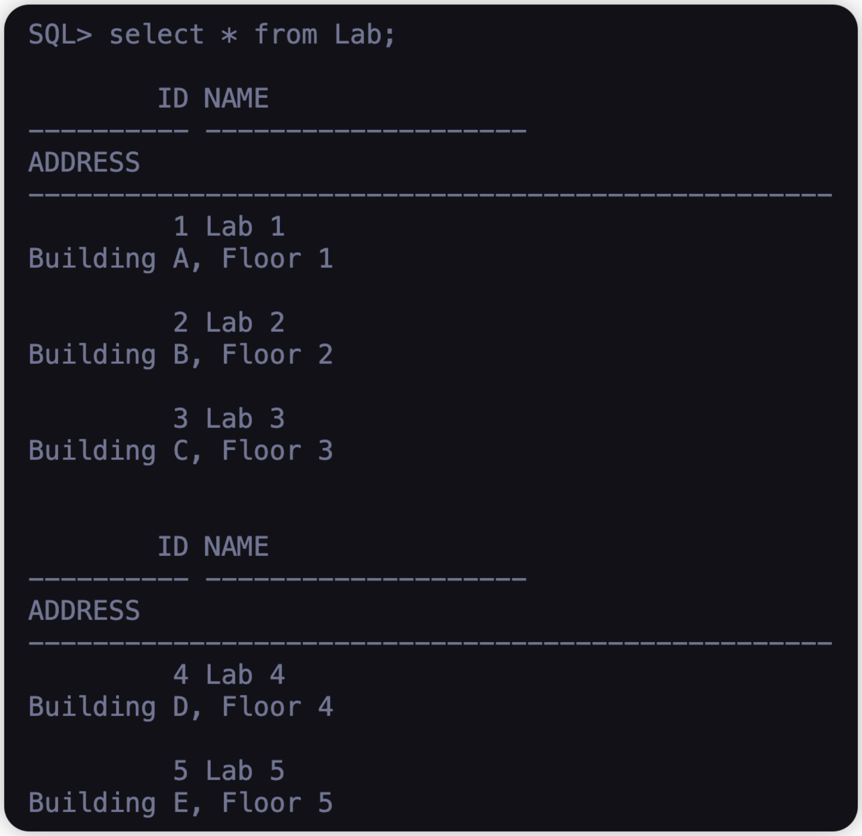
```
SQL> select * from LabMembers;

USERID               NAME
-------------------- --------------------
EMAIL                                              PHONE
-------------------------------------------------- --------------------
user1                John Smith
john.smith@example.com                             123-456-7890

user2                Sam Doe
sam.doe@example.com                                234-567-8901

user3                Robert Johnson
robert.johnson@example.com                         345-678-9012


USERID               NAME
-------------------- --------------------
EMAIL                                              PHONE
-------------------------------------------------- --------------------
user4                Emily Wilson
emily.wilson@example.com                           456-789-0123

user5                Michael Brown
michael.brown@example.com                          567-890-1234
```

- Use (**<u>CatalogNumber</u>**, **<u>UserID</u>**, UseDate)



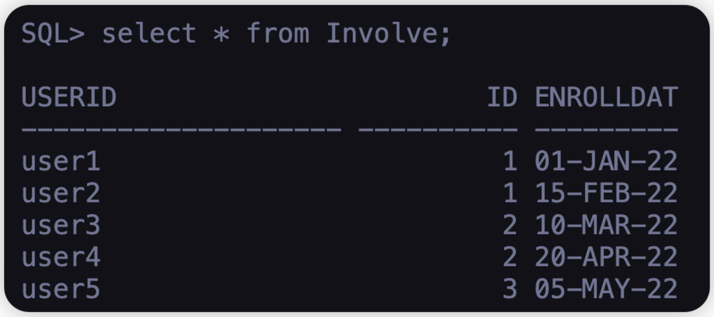
```
SQL> select * from Use;

CATALOGNUMBER USERID                    USEDATE
------------- -------------------- ---------
         1001 user1                     31-JAN-23
         1002 user2                     03-FEB-23
         1003 user3                     30-APR-23
         1004 user4                     31-MAY-23
         1005 user5                     28-MAR-23
```

- Lab (<u>ID</u>, Name, Address)



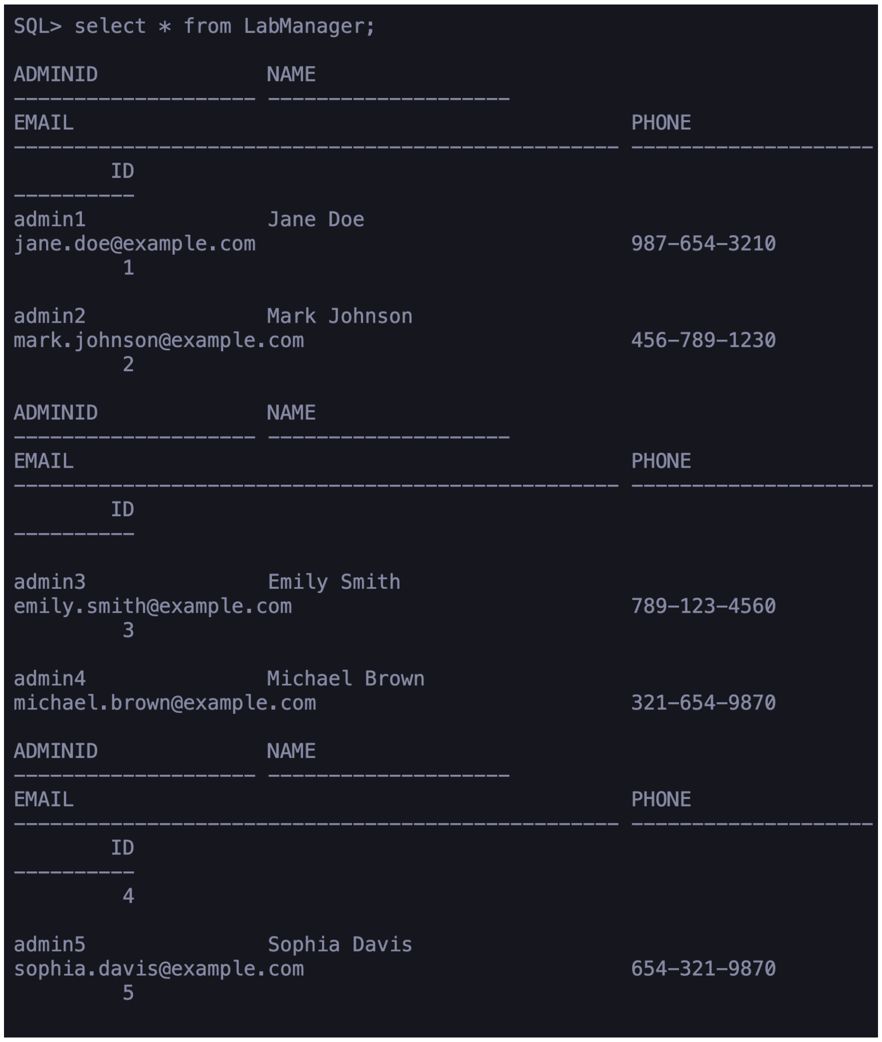
```
SQL> select * from Lab;

        ID NAME
---------- --------------------
ADDRESS
--------------------------------------------------
         1 Lab 1
Building A, Floor 1

         2 Lab 2
Building B, Floor 2

         3 Lab 3
Building C, Floor 3


        ID NAME
---------- --------------------
ADDRESS
--------------------------------------------------
         4 Lab 4
Building D, Floor 4

         5 Lab 5
Building E, Floor 5
```

- Involve (<u>**UserID**</u>, <u>**ID**</u>, EnrollDate)



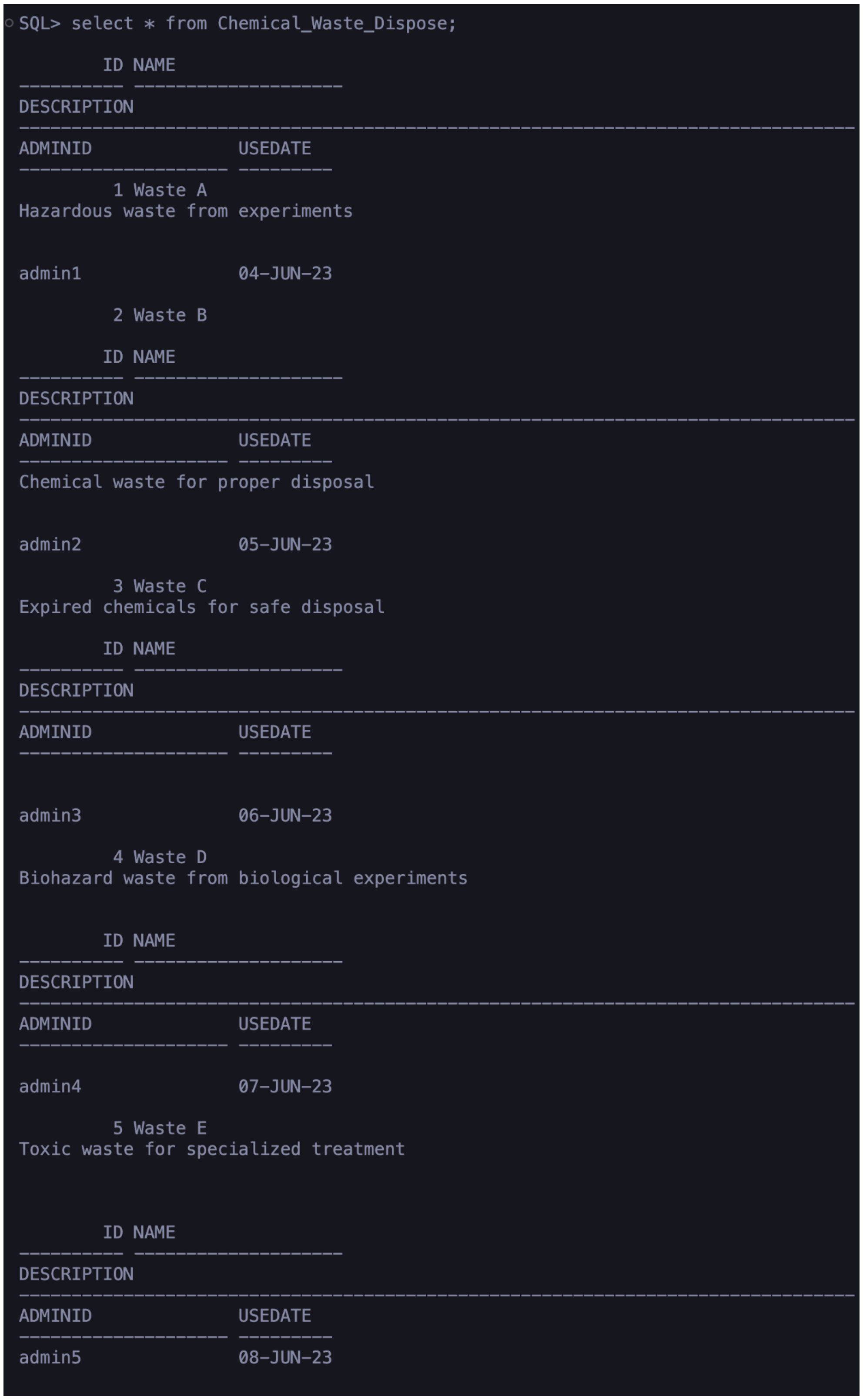
```
SQL> select * from Involve;

USERID                                ID ENROLLDAT
-------------------- ---------- ---------
user1                                  1 01-JAN-22
user2                                  1 15-FEB-22
user3                                  2 10-MAR-22
user4                                  2 20-APR-22
user5                                  3 05-MAY-22
```

- LabManager (<u>AdminID</u>, Name, Email, Phone, **ID**)



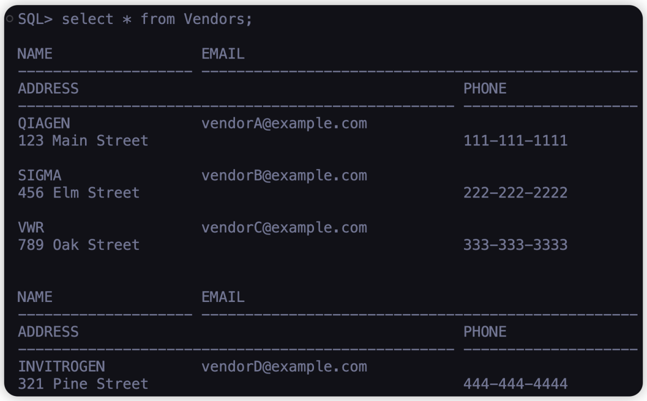
```
SQL> select * from LabManager;

ADMINID              NAME
-------------------- --------------------
EMAIL                                              PHONE
-------------------------------------------------- --------------------
        ID
----------
admin1               Jane Doe
jane.doe@example.com                               987-654-3210
         1

admin2               Mark Johnson
mark.johnson@example.com                           456-789-1230
         2

ADMINID              NAME
-------------------- --------------------
EMAIL                                              PHONE
-------------------------------------------------- --------------------
        ID
----------

admin3               Emily Smith
emily.smith@example.com                            789-123-4560
         3

admin4               Michael Brown
michael.brown@example.com                          321-654-9870

ADMINID              NAME
-------------------- --------------------
EMAIL                                              PHONE
-------------------------------------------------- --------------------
        ID
----------
         4

admin5               Sophia Davis
sophia.davis@example.com                           654-321-9870
         5
```

- Chemical_Waste_Dispose(<u>ID</u>, Name, Description, **AdminID**, UseDate)



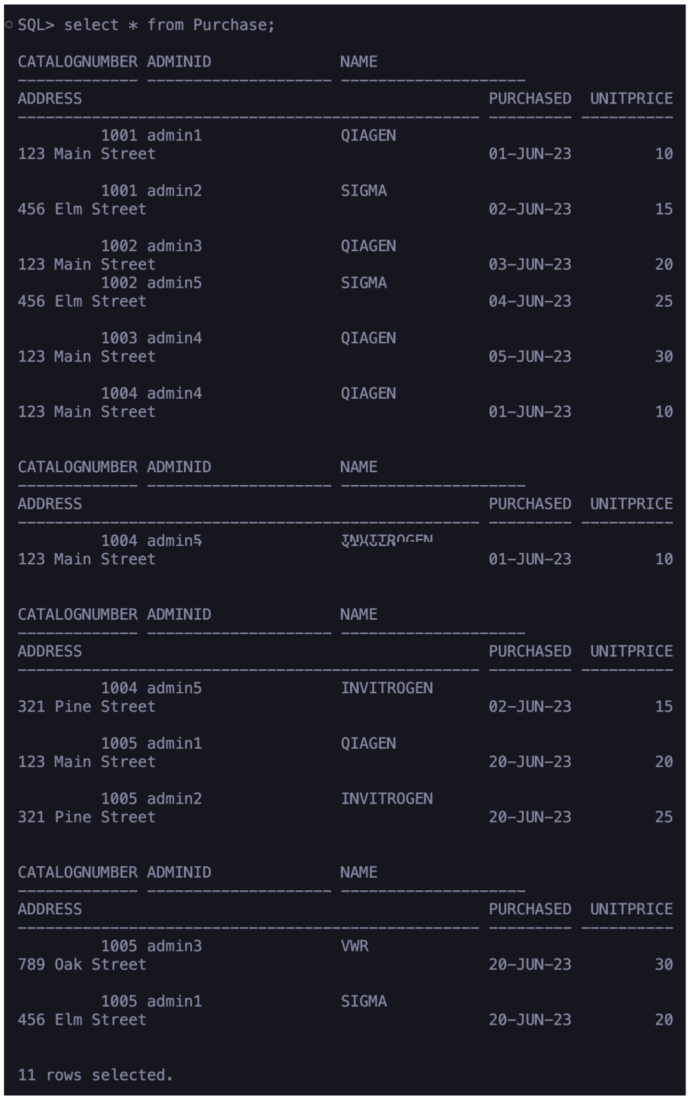
```
SQL> select * from Chemical_Waste_Dispose;

        ID NAME
---------- --------------------
DESCRIPTION
--------------------------------------------------------------------------------
ADMINID              USEDATE
-------------------- ---------
         1 Waste A
Hazardous waste from experiments


admin1               04-JUN-23

         2 Waste B

        ID NAME
---------- --------------------
DESCRIPTION
--------------------------------------------------------------------------------
ADMINID              USEDATE
-------------------- ---------
Chemical waste for proper disposal


admin2               05-JUN-23

         3 Waste C
Expired chemicals for safe disposal

        ID NAME
---------- --------------------
DESCRIPTION
--------------------------------------------------------------------------------
ADMINID              USEDATE
-------------------- ---------


admin3               06-JUN-23

         4 Waste D
Biohazard waste from biological experiments


        ID NAME
---------- --------------------
DESCRIPTION
--------------------------------------------------------------------------------
ADMINID              USEDATE
-------------------- ---------

admin4               07-JUN-23

         5 Waste E
Toxic waste for specialized treatment



        ID NAME
---------- --------------------
DESCRIPTION
--------------------------------------------------------------------------------
ADMINID              USEDATE
-------------------- ---------
admin5               08-JUN-23
```

- Vendors (<u>Name</u>, Email, <u>Address</u>, Phone)



```
SQL> select * from Vendors;

NAME                 EMAIL
-------------------- --------------------------------------------------
ADDRESS                                            PHONE
-------------------------------------------------- --------------------
QIAGEN               vendorA@example.com
123 Main Street                                    111-111-1111

SIGMA                vendorB@example.com
456 Elm Street                                     222-222-2222

VWR                  vendorC@example.com
789 Oak Street                                     333-333-3333


NAME                 EMAIL
-------------------- --------------------------------------------------
ADDRESS                                            PHONE
-------------------------------------------------- --------------------
INVITROGEN           vendorD@example.com
321 Pine Street                                    444-444-4444
```

- Purchase (**<u>CatalogNumber</u>**, **<u>AdminID</u>**, **<u>Name</u>**, **<u>Address</u>**, PurchaseDate, UnitPrice)



```
SQL> select * from Purchase;

CATALOGNUMBER ADMINID              NAME
------------- -------------------- --------------------
ADDRESS                                            PURCHASED  UNITPRICE
-------------------------------------------------- --------- ----------
         1001 admin1               QIAGEN
123 Main Street                                    01-JUN-23         10

         1001 admin2               SIGMA
456 Elm Street                                     02-JUN-23         15

         1002 admin3               QIAGEN
123 Main Street                                    03-JUN-23         20
         1002 admin5               SIGMA
456 Elm Street                                     04-JUN-23         25

         1003 admin4               QIAGEN
123 Main Street                                    05-JUN-23         30

         1004 admin4               QIAGEN
123 Main Street                                    01-JUN-23         10


CATALOGNUMBER ADMINID              NAME
------------- -------------------- --------------------
ADDRESS                                            PURCHASED  UNITPRICE
-------------------------------------------------- --------- ----------
         1004 admin5               INVITROGEN
123 Main Street                                    01-JUN-23         10


CATALOGNUMBER ADMINID              NAME
------------- -------------------- --------------------
ADDRESS                                            PURCHASED  UNITPRICE
-------------------------------------------------- --------- ----------
         1004 admin5               INVITROGEN
321 Pine Street                                    02-JUN-23         15

         1005 admin1               QIAGEN
123 Main Street                                    20-JUN-23         20

         1005 admin2               INVITROGEN
321 Pine Street                                    20-JUN-23         25


CATALOGNUMBER ADMINID              NAME
------------- -------------------- --------------------
ADDRESS                                            PURCHASED  UNITPRICE
-------------------------------------------------- --------- ----------
         1005 admin3               VWR
789 Oak Street                                     20-JUN-23         30

         1005 admin1               SIGMA
456 Elm Street                                     20-JUN-23         20


11 rows selected.
```

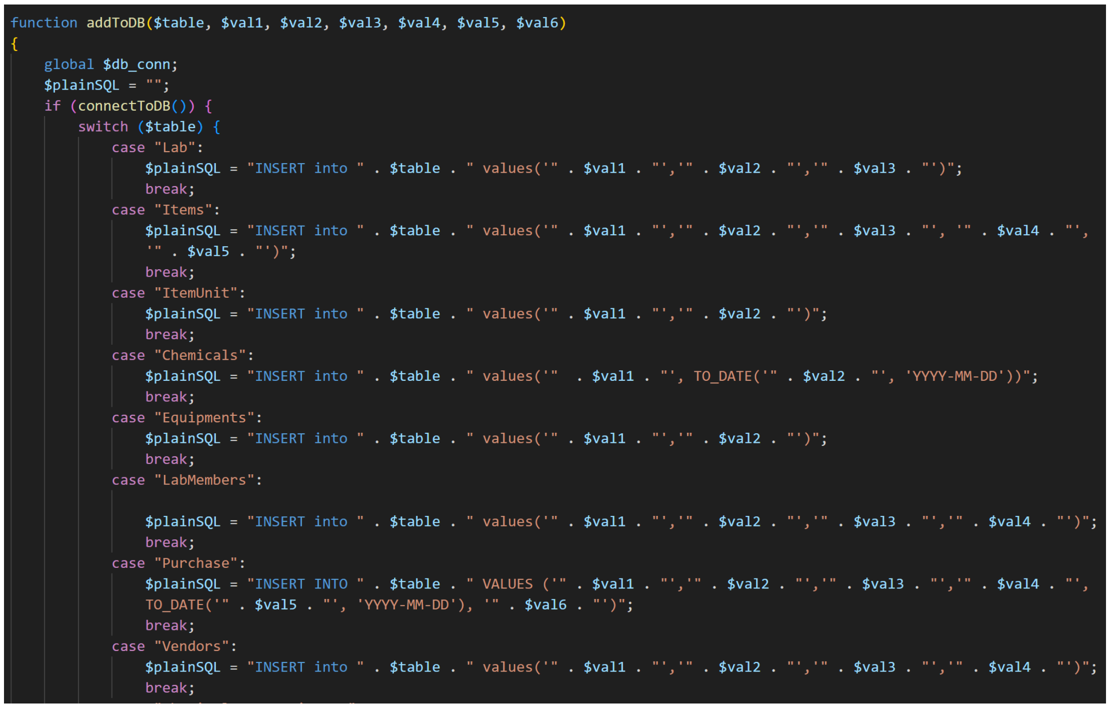
4. **A list of all SQL queries used. For SQL query requirements, check the rubric listed on Canvas for Milestone 4.**

   Please refer to Question 5.

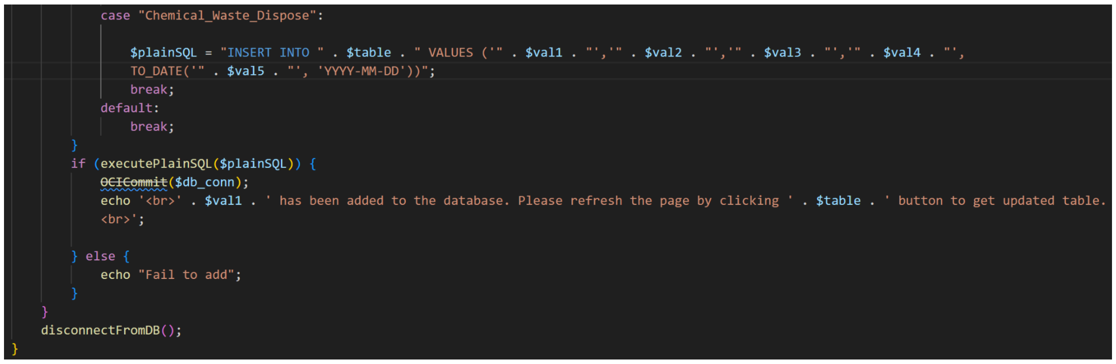
5. **Screenshots of the sample output of the queries using the GUI (for example, you can show what data is in your table before you run the query, and then show another screenshot after running the query, from some kind of GUI input like a button).**

SQL Query and GUI OUTPUT 👍

Insert



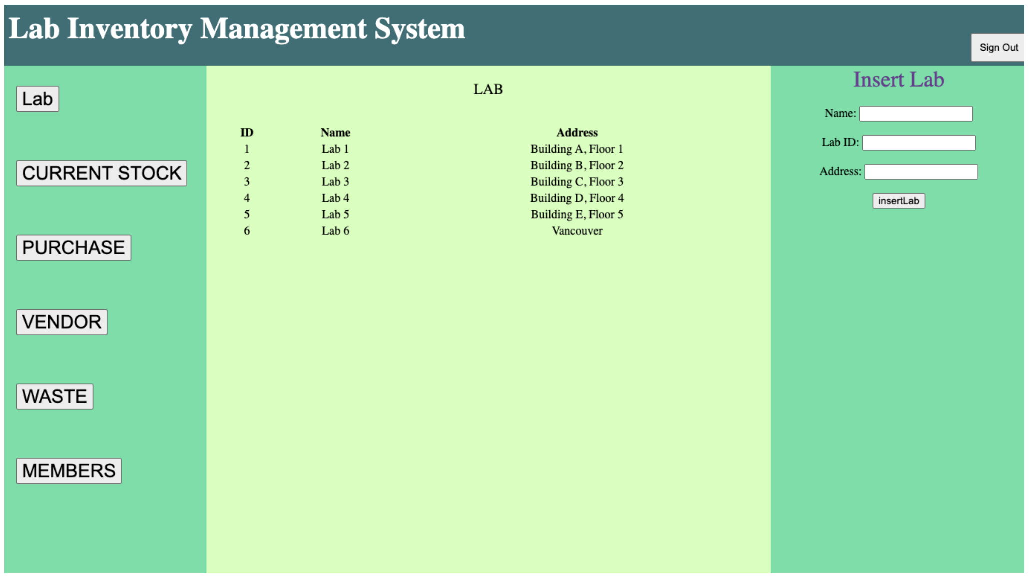
```php
function addToDB($table, $val1, $val2, $val3, $val4, $val5, $val6)
{
    global $db_conn;
    $plainSQL = "";
    if (connectToDB()) {
        switch ($table) {
            case "Lab":
                $plainSQL = "INSERT into " . $table . " values('" . $val1 . "','" . $val2 . "','" . $val3 . "')";
                break;
            case "Items":
                $plainSQL = "INSERT into " . $table . " values('" . $val1 . "','" . $val2 . "','" . $val3 . "', '" . $val4 . "',
                '" . $val5 . "')";
                break;
            case "ItemUnit":
                $plainSQL = "INSERT into " . $table . " values('" . $val1 . "','" . $val2 . "')";
                break;
            case "Chemicals":
                $plainSQL = "INSERT into " . $table . " values('"  . $val1 . "', TO_DATE('" . $val2 . "', 'YYYY-MM-DD'))";
                break;
            case "Equipments":
                $plainSQL = "INSERT into " . $table . " values('" . $val1 . "','" . $val2 . "')";
                break;
            case "LabMembers":

                $plainSQL = "INSERT into " . $table . " values('" . $val1 . "','" . $val2 . "','" . $val3 . "','" . $val4 . "')";
                break;
            case "Purchase":
                $plainSQL = "INSERT INTO " . $table . " VALUES ('" . $val1 . "','" . $val2 . "','" . $val3 . "','" . $val4 . "',
                TO_DATE('" . $val5 . "', 'YYYY-MM-DD'), '" . $val6 . "')";
                break;
            case "Vendors":
                $plainSQL = "INSERT into " . $table . " values('" . $val1 . "','" . $val2 . "','" . $val3 . "','" . $val4 . "')";
                break;
```



```php
            case "Chemical_Waste_Dispose":

                $plainSQL = "INSERT INTO " . $table . " VALUES ('" . $val1 . "','" . $val2 . "','" . $val3 . "','" . $val4 . "',
                TO_DATE('" . $val5 . "', 'YYYY-MM-DD'))";
                break;
            default:
                break;
        }
        if (executePlainSQL($plainSQL)) {
            OCICommit($db_conn);
            echo '<br>' . $val1 . ' has been added to the database. Please refresh the page by clicking ' . $table . ' button to get updated table.
            <br>';

        } else {
            echo "Fail to add";
        }
    }
    disconnectFromDB();
}
```

University of British Columbia, Vancouver
Department of Computer Science

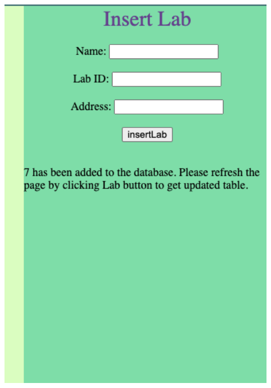
Before Insert



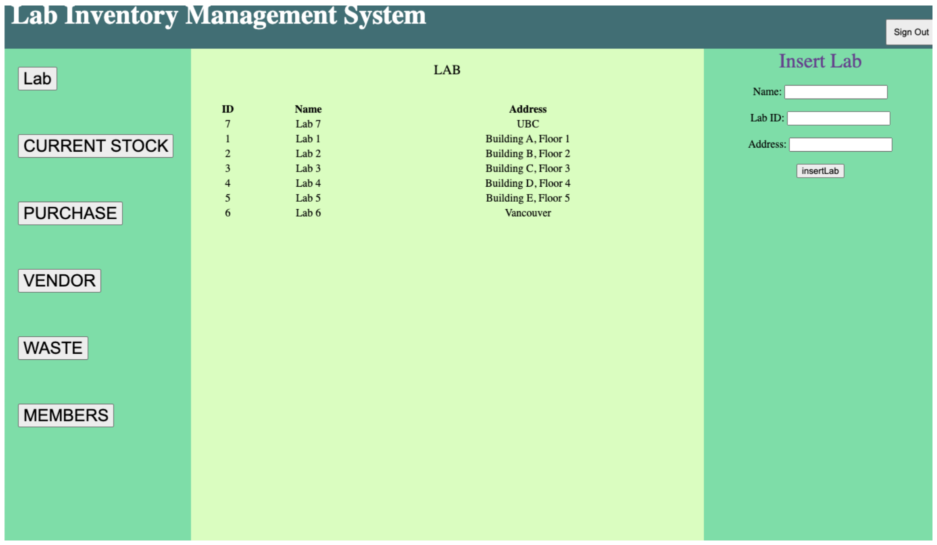
LAB

ID	Name	Address
1	Lab 1	Building A, Floor 1
2	Lab 2	Building B, Floor 2
3	Lab 3	Building C, Floor 3
4	Lab 4	Building D, Floor 4
5	Lab 5	Building E, Floor 5
6	Lab 6	Vancouver

After Insert



7 has been added to the database. Please refresh the page by clicking Lab button to get updated table.

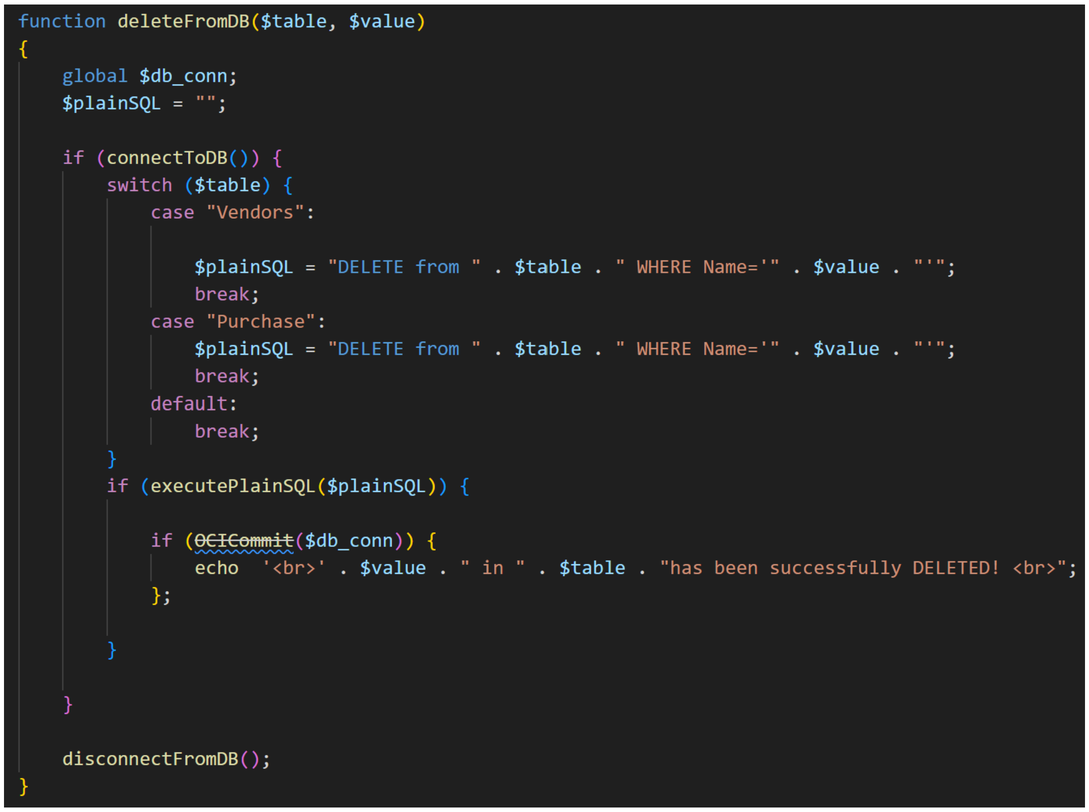


LAB

ID	Name	Address
7	Lab 7	UBC
1	Lab 1	Building A, Floor 1
2	Lab 2	Building B, Floor 2
3	Lab 3	Building C, Floor 3
4	Lab 4	Building D, Floor 4
5	Lab 5	Building E, Floor 5
6	Lab 6	Vancouver

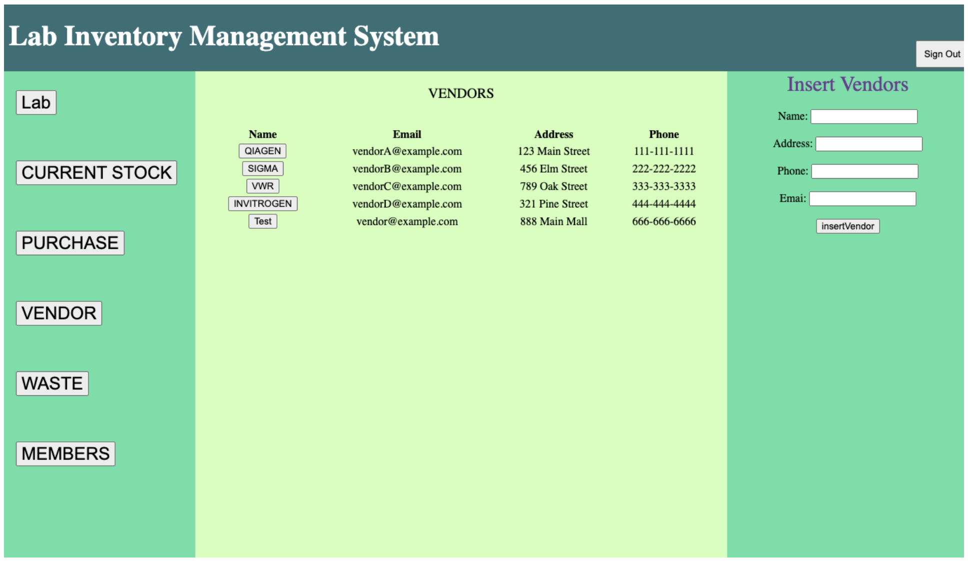
University of British Columbia, Vancouver
Department of Computer Science

Delete



```
function deleteFromDB($table, $value)
{
    global $db_conn;
    $plainSQL = "";

    if (connectToDB()) {
        switch ($table) {
            case "Vendors":

                $plainSQL = "DELETE from " . $table . " WHERE Name='" . $value . "'";
                break;
            case "Purchase":
                $plainSQL = "DELETE from " . $table . " WHERE Name='" . $value . "'";
                break;
            default:
                break;
        }
        if (executePlainSQL($plainSQL)) {

            if (~~OCICommit~~($db_conn)) {
                echo  '<br>' . $value . " in " . $table . "has been successfully DELETED! <br>";
            };

        }

    }

    disconnectFromDB();
}
```

Before Delete



Lab Inventory Management System

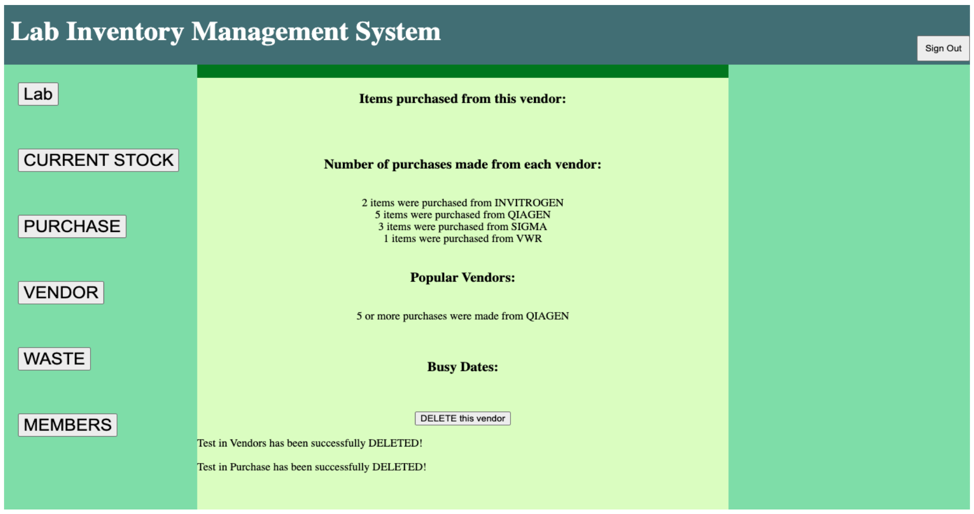
VENDORS

Name	Email	Address	Phone
QIAGEN	vendorA@example.com	123 Main Street	111-111-1111
SIGMA	vendorB@example.com	456 Elm Street	222-222-2222
VWR	vendorC@example.com	789 Oak Street	333-333-3333
INVITROGEN	vendorD@example.com	321 Pine Street	444-444-4444
Test	vendor@example.com	888 Main Mall	666-666-6666

Insert Vendors

Name:
Address:
Phone:
Emai:

After Delete



Items purchased from this vendor:

Number of purchases made from each vendor:

2 items were purchased from INVITROGEN
5 items were purchased from QIAGEN
3 items were purchased from SIGMA
1 items were purchased from VWR

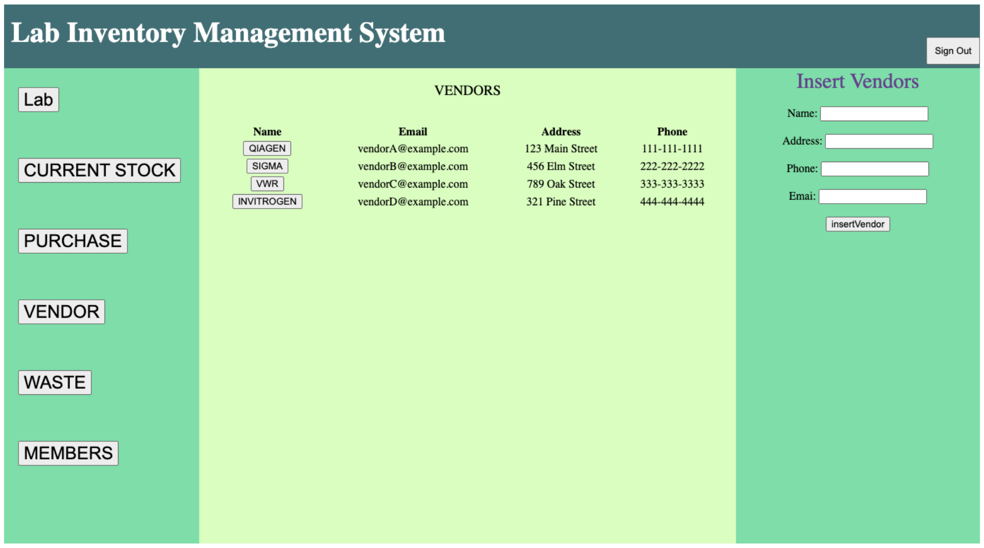
Popular Vendors:

5 or more purchases were made from QIAGEN

Busy Dates:

Test in Vendors has been successfully DELETED!

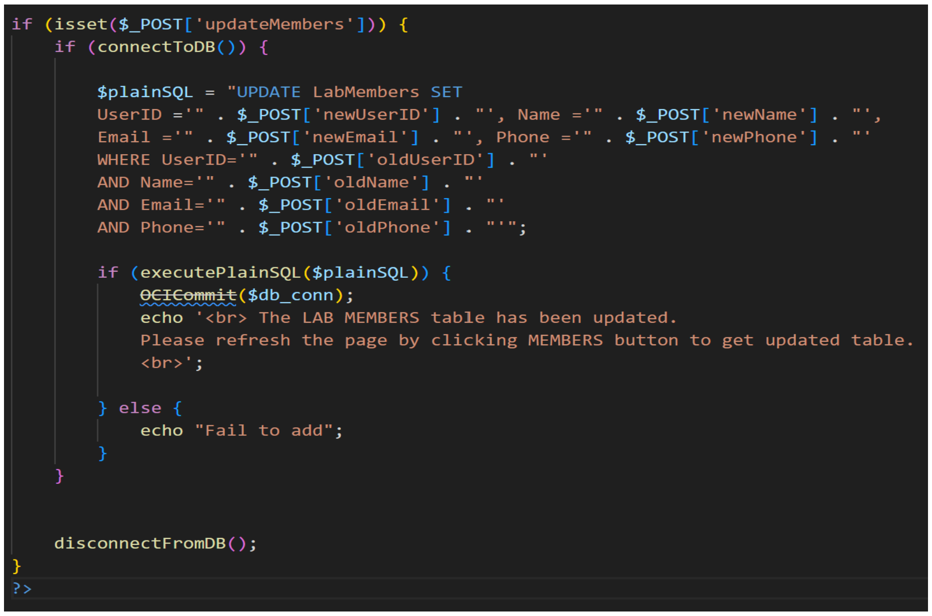
Test in Purchase has been successfully DELETED!



VENDORS

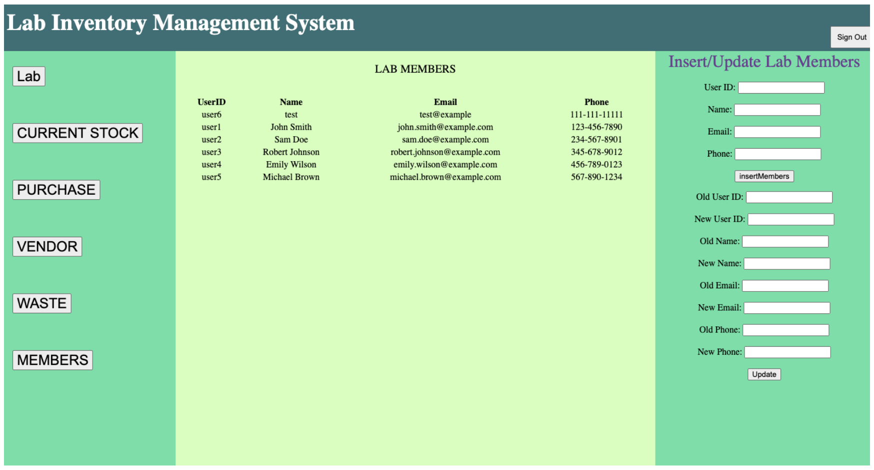
Name	Email	Address	Phone
QIAGEN	vendorA@example.com	123 Main Street	111-111-1111
SIGMA	vendorB@example.com	456 Elm Street	222-222-2222
VWR	vendorC@example.com	789 Oak Street	333-333-3333
INVITROGEN	vendorD@example.com	321 Pine Street	444-444-4444

Update



```
if (isset($_POST['updateMembers'])) {
    if (connectToDB()) {

        $plainSQL = "UPDATE LabMembers SET
        UserID ='" . $_POST['newUserID'] . "', Name ='" . $_POST['newName'] . "',
        Email ='" . $_POST['newEmail'] . "', Phone ='" . $_POST['newPhone'] . "'
        WHERE UserID='" . $_POST['oldUserID'] . "'
        AND Name='" . $_POST['oldName'] . "'
        AND Email='" . $_POST['oldEmail'] . "'
        AND Phone='" . $_POST['oldPhone'] . "'";

        if (executePlainSQL($plainSQL)) {
            ~~OCICommit~~($db_conn);
            echo '<br> The LAB MEMBERS table has been updated.
            Please refresh the page by clicking MEMBERS button to get updated table.
            <br>';

        } else {
            echo "Fail to add";
        }
    }


    disconnectFromDB();
}
?>
```

Before Update



Lab Inventory Management System

LAB MEMBERS

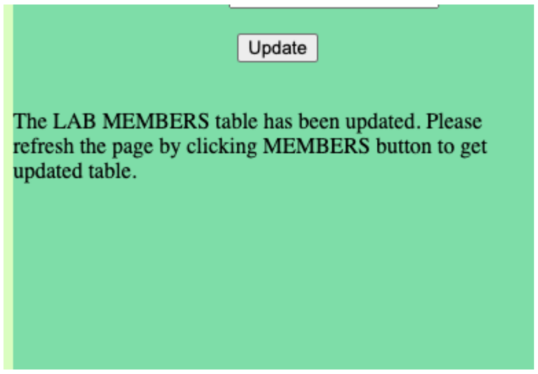
UserID	Name	Email	Phone
user6	test	test@example	111-111-11111
user1	John Smith	john.smith@example.com	123-456-7890
user2	Sam Doe	sam.doe@example.com	234-567-8901
user3	Robert Johnson	robert.johnson@example.com	345-678-9012
user4	Emily Wilson	emily.wilson@example.com	456-789-0123
user5	Michael Brown	michael.brown@example.com	567-890-1234

Insert/Update Lab Members

User ID:
Name:
Email:
Phone:

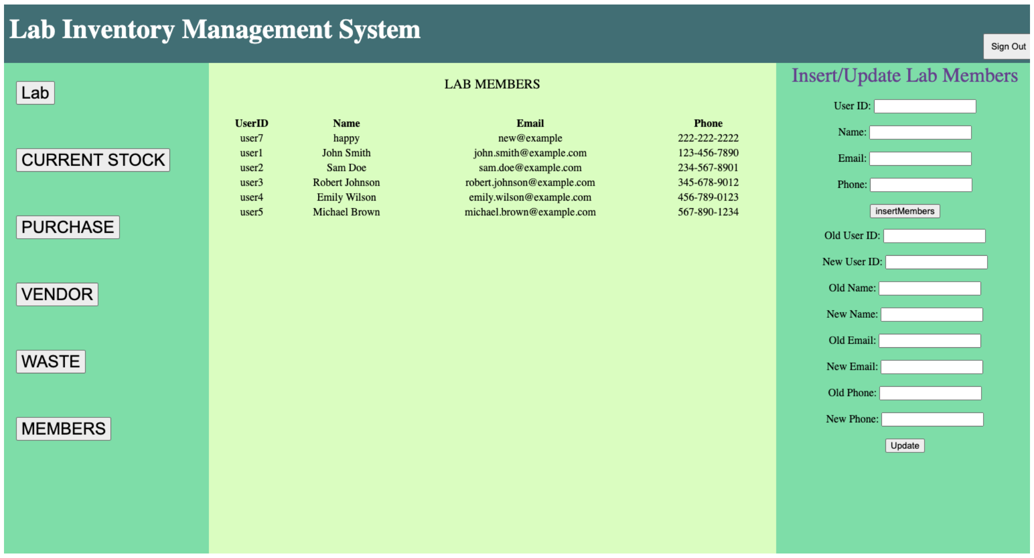
Old User ID:
New User ID:
Old Name:
New Name:
Old Email:
New Email:
Old Phone:
New Phone:

After Update



The LAB MEMBERS table has been updated. Please refresh the page by clicking MEMBERS button to get updated table.

University of British Columbia, Vancouver
Department of Computer Science



Lab Inventory Management System

Sign Out

Lab

CURRENT STOCK

PURCHASE

VENDOR

WASTE

MEMBERS

LAB MEMBERS

UserID	Name	Email	Phone
user7	happy	new@example	222-222-2222
user1	John Smith	john.smith@example.com	123-456-7890
user2	Sam Doe	sam.doe@example.com	234-567-8901
user3	Robert Johnson	robert.johnson@example.com	345-678-9012
user4	Emily Wilson	emily.wilson@example.com	456-789-0123
user5	Michael Brown	michael.brown@example.com	567-890-1234

Insert/Update Lab Members

User ID:

Name:

Email:

Phone:

insertMembers

Old User ID:

New User ID:

Old Name:

New Name:

Old Email:

New Email:

Old Phone:

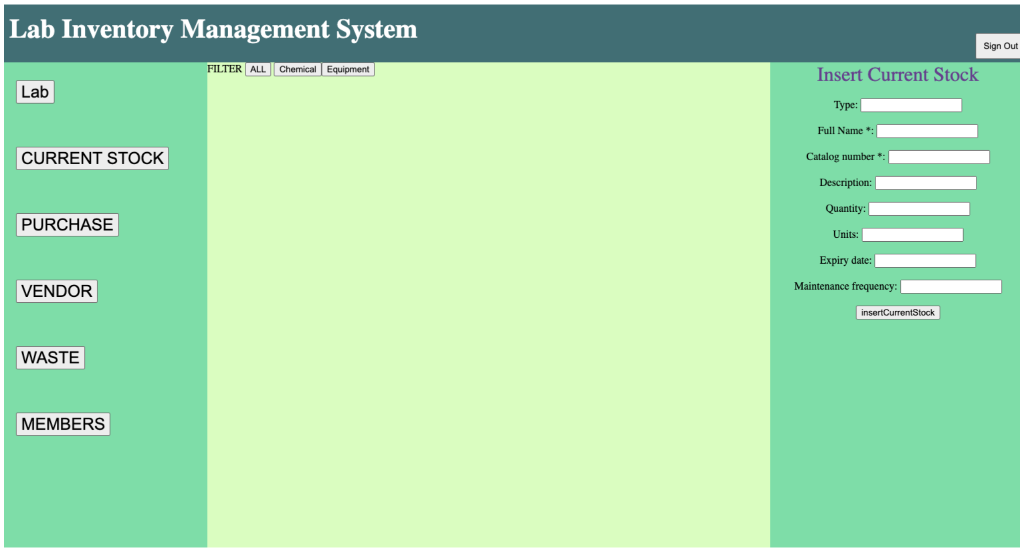
New Phone:

Update

Selection

```
$result = executePlainSQL("SELECT * FROM " . $table);
```

Before Selection



Lab Inventory Management System

Sign Out

FILTER ALL Chemical Equipment

Lab

CURRENT STOCK

PURCHASE

VENDOR

WASTE

MEMBERS

Insert Current Stock

Type:

Full Name *:

Catalog number *:

Description:

Quantity:

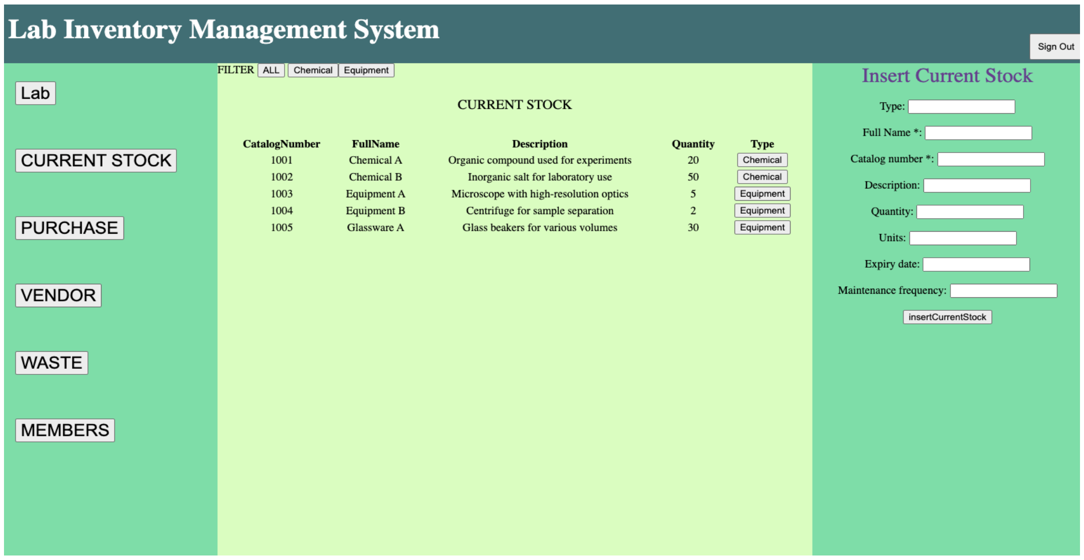
Units:

Expiry date:

Maintenance frequency:

insertCurrentStock

After Selection



Lab Inventory Management System

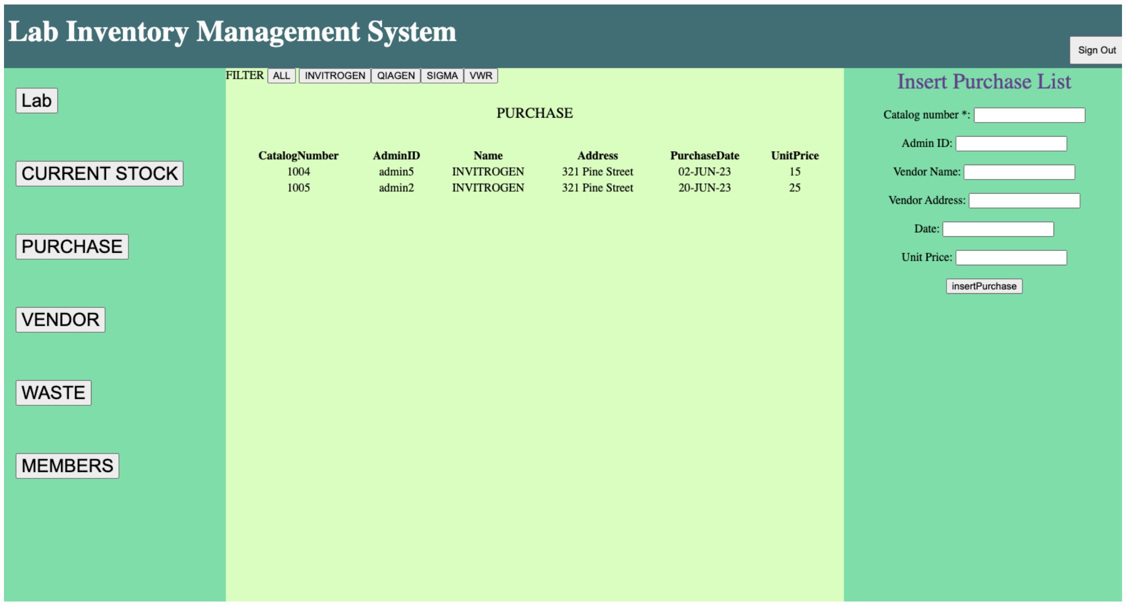
CURRENT STOCK

CatalogNumber	FullName	Description	Quantity	Type
1001	Chemical A	Organic compound used for experiments	20	Chemical
1002	Chemical B	Inorganic salt for laboratory use	50	Chemical
1003	Equipment A	Microscope with high-resolution optics	5	Equipment
1004	Equipment B	Centrifuge for sample separation	2	Equipment
1005	Glassware A	Glass beakers for various volumes	30	Equipment

Projection

```
    $result = executePlainSQL("SELECT * FROM " . $table . " WHERE " . $mode . "='" . $value . "'");
}
```

Below is one of examples that we did projection for only Vendors Name = “INVITROGEN”



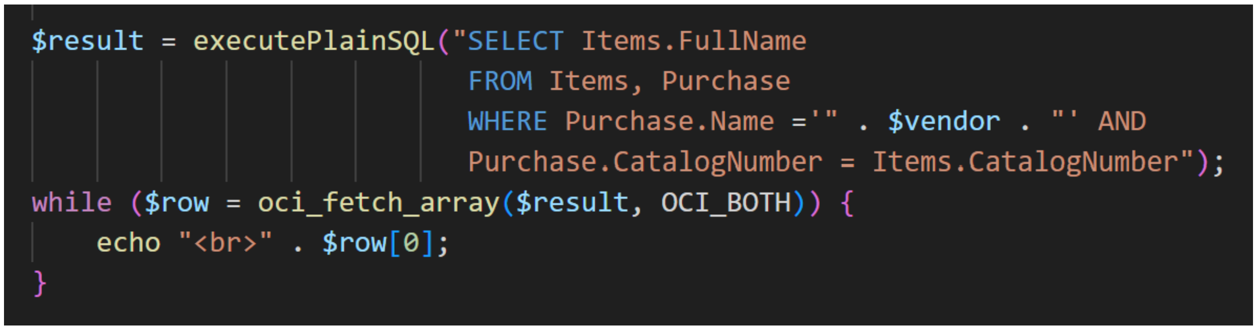
Lab Inventory Management System

PURCHASE

CatalogNumber	AdminID	Name	Address	PurchaseDate	UnitPrice
1004	admin5	INVITROGEN	321 Pine Street	02-JUN-23	15
1005	admin2	INVITROGEN	321 Pine Street	20-JUN-23	25

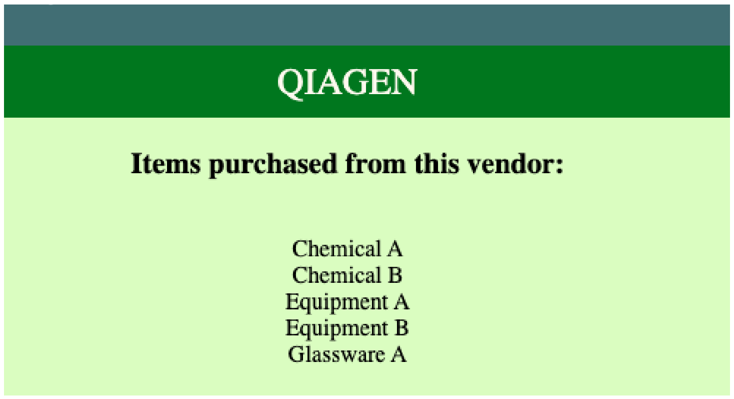
University of British Columbia, Vancouver
Department of Computer Science

Join



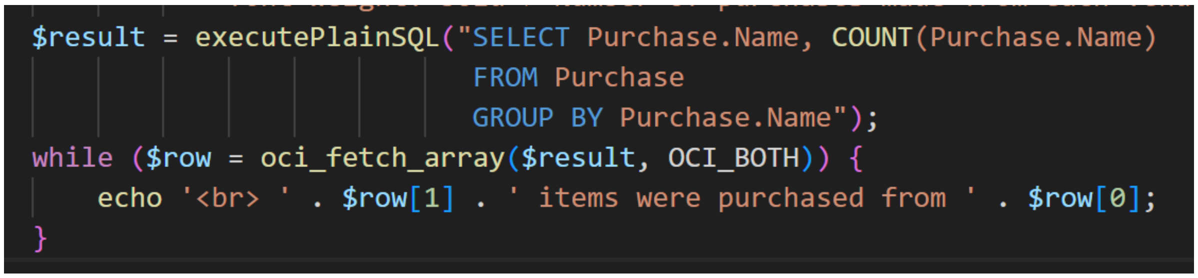
```php
$result = executePlainSQL("SELECT Items.FullName
                           FROM Items, Purchase
                           WHERE Purchase.Name ='" . $vendor . "' AND
                           Purchase.CatalogNumber = Items.CatalogNumber");
while ($row = oci_fetch_array($result, OCI_BOTH)) {
    echo "<br>" . $row[0];
}
```

An example: Join shows all Items Names Purchased from the Vendor named “Qiagen”



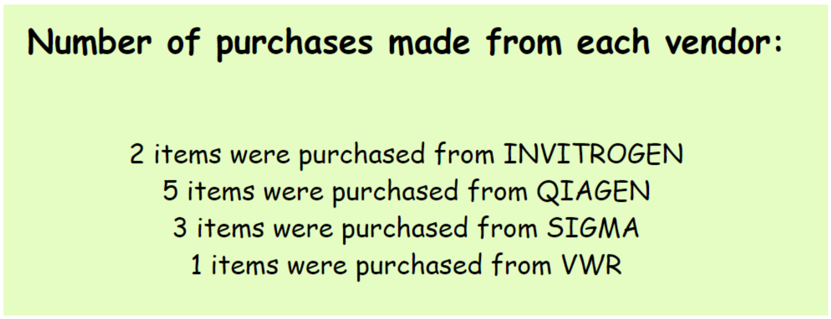
```
QIAGEN

Items purchased from this vendor:

Chemical A
Chemical B
Equipment A
Equipment B
Glassware A
```

Aggregation with Group by



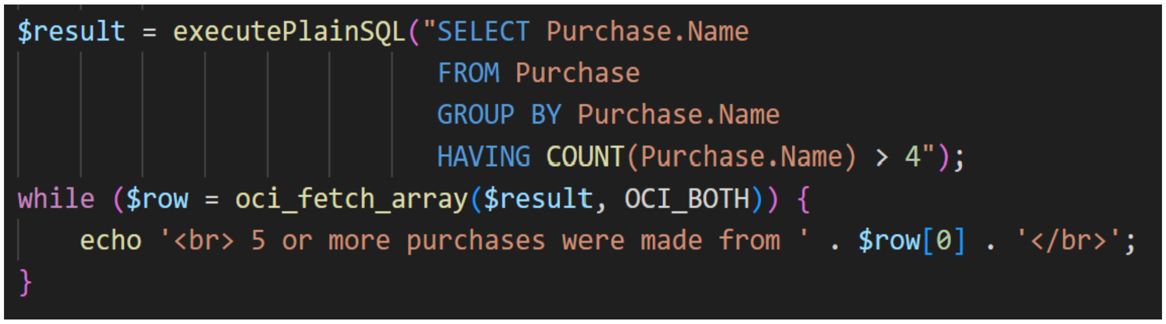
```php
$result = executePlainSQL("SELECT Purchase.Name, COUNT(Purchase.Name)
                           FROM Purchase
                           GROUP BY Purchase.Name");
while ($row = oci_fetch_array($result, OCI_BOTH)) {
    echo '<br> ' . $row[1] . ' items were purchased from ' . $row[0];
}
```

An example: count the number of purchases from each vendor



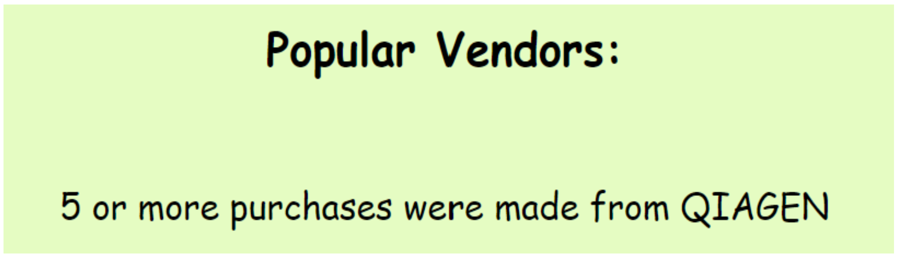
```
Number of purchases made from each vendor:

2 items were purchased from INVITROGEN
5 items were purchased from QIAGEN
3 items were purchased from SIGMA
1 items were purchased from VWR
```

Aggregation with Having



```
$result = executePlainSQL("SELECT Purchase.Name
                           FROM Purchase
                           GROUP BY Purchase.Name
                           HAVING COUNT(Purchase.Name) > 4");
while ($row = oci_fetch_array($result, OCI_BOTH)) {
    echo '<br> 5 or more purchases were made from ' . $row[0] . '</br>';
}
```

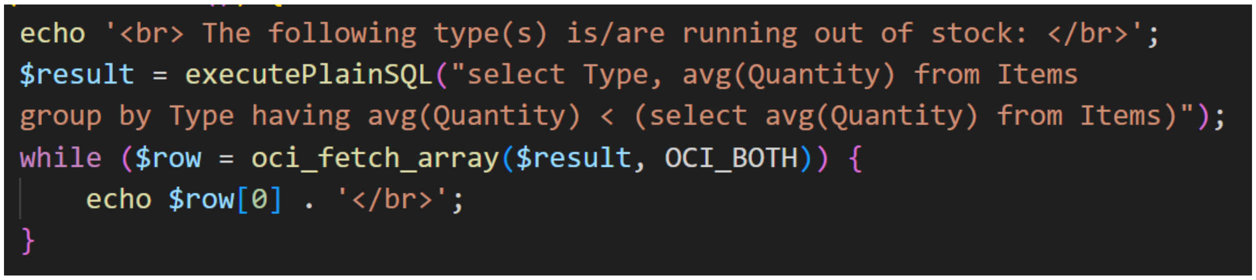
An example: The vendor's name where 5 or more purchases were made from.



Popular Vendors:

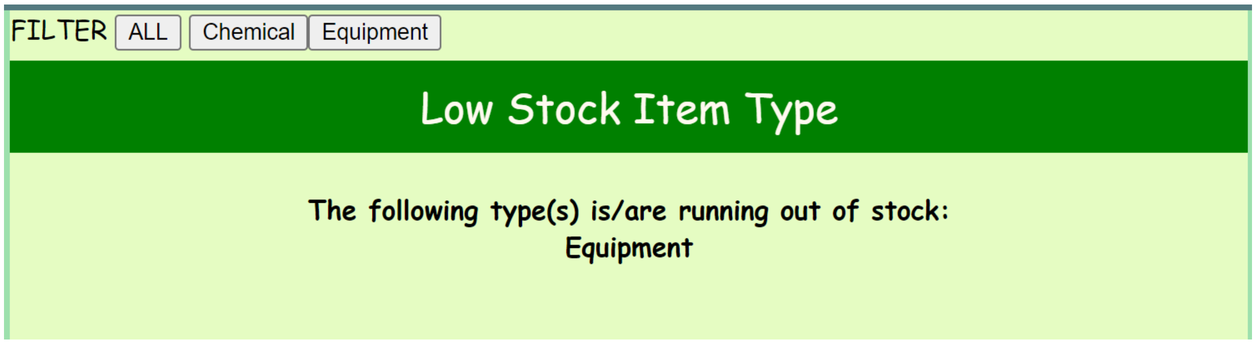
5 or more purchases were made from QIAGEN

Nested Aggregation with Group by



```
echo '<br> The following type(s) is/are running out of stock: </br>';
$result = executePlainSQL("select Type, avg(Quantity) from Items
group by Type having avg(Quantity) < (select avg(Quantity) from Items)");
while ($row = oci_fetch_array($result, OCI_BOTH)) {
    echo $row[0] . '</br>';
}
```

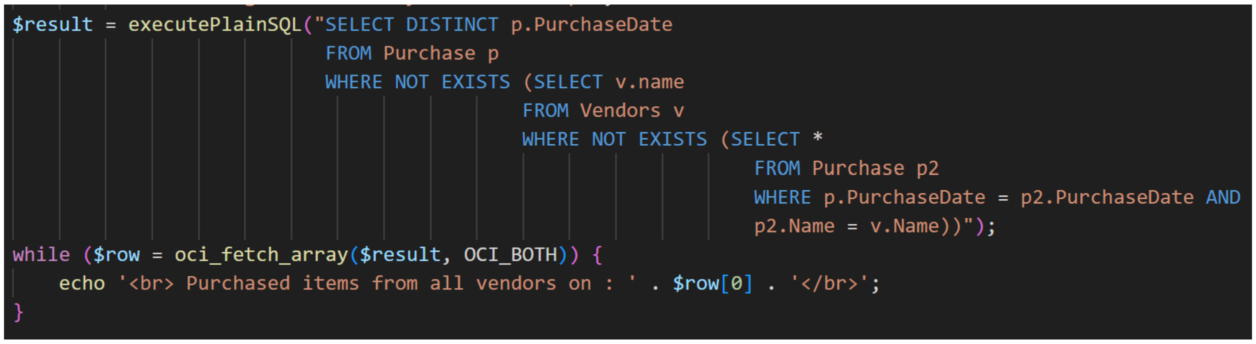
An example: The type of items that are low on stock (less than the average quantity)



Low Stock Item Type

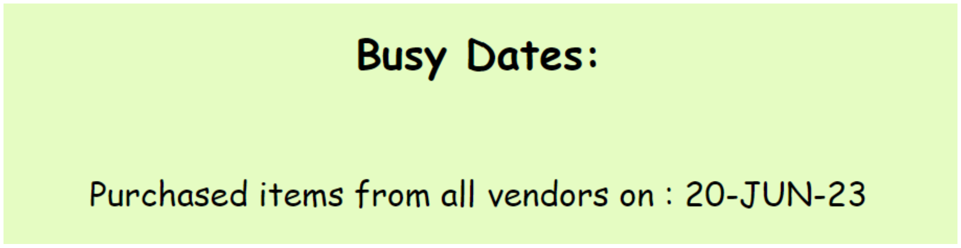
The following type(s) is/are running out of stock:
Equipment

Division



```
$result = executePlainSQL("SELECT DISTINCT p.PurchaseDate
                           FROM Purchase p
                           WHERE NOT EXISTS (SELECT v.name
                                             FROM Vendors v
                                             WHERE NOT EXISTS (SELECT *
                                                               FROM Purchase p2
                                                               WHERE p.PurchaseDate = p2.PurchaseDate AND
                                                               p2.Name = v.Name))");
while ($row = oci_fetch_array($result, OCI_BOTH)) {
    echo '<br> Purchased items from all vendors on : ' . $row[0] . '</br>';
}
```

An example: Displays items purchased by ALL vendors



Busy Dates:

Purchased items from all vendors on : 20-JUN-23