Lab #1

- Look at COMPANY schema
- login to local PHPMyAdmin
  - Run Wamp/MampServer
  - Open PHPMyAdmin
- Click on NEW in left menu bar
- click on SQL tab on top
  - ➔ CREATE DATABASE COMPANY;
- See the company DB in the left
- Open the CompanyDDL.txt file on your local computer [CompanyDDL.]
  (there is a link on the syllabus and on Sakai)
  (use Notepad++ or TextWranger or similar: it's a plain text file)
- Select and copy the entire file
- In PHPMyAdmin, paste into the SQL window.
  Click GO
- Note that the CompanyDDL.txt included both the DDL and the Insert
- Repeat the above instructions and create CompanyBackup DB
  - You can copy a DB on the command line: mysqlDBcopy
  - Alternatively, also on the command line: mysqlDump
  - -->You can also copy or rename a DB in PHPMYAdmin copy/rename
- In SQL window, type SHOW TABLE STATUS; and discuss INNODB vs. MyISAM
  (foreign key support; transactions; row-level locking)

That was the warmup! Now for the main event.
Look at the DDL for Pine Valley.

1. Display all fields in the table Customer_t.

2. Display product name and quantity on hand from PRODUCT
   for all Pine Valley Furniture Company products that have a std price < $275

3. What is the address of the customer named Home Furnishings?
   Use an alias, Name for the Customer Name.

NOTE: Due to the order of operations, you can't use "Name" (the alias) in the WHER
But it will change the column header to Name.

4. List the unit price, product name, and product ID for all products in the Product

5. What are the standard price and standard price if increased by 10% for every p

6. What is the average standard price for all products in inventory?

7 How many different items were ordered on Order number 1004?

8. How many different items were ordered on order number 1004, and what are t

```
SELECT PRoductID, COUNT(*)
FROM OrderLine_t
WHERE OrderID=1004;
```

Note that the result shown is 6 2. ***This is NOT correct!! (scalar/aggreg***

Look at the OrderLine table. There are two ProductIDs and a count of 2.
It retrieved the first ProductID and the count of 2.
In many versions of SQL, this would generate an error.

9. Find the difference between the std price of each product and the average price of all

```
SELECT ProductStandardPrice -AVG(ProductStandardPrice) FROM Product_t;  <-
```

This results in -265.625. This is clearly incorrect. Again, mixing scalar and aggre

10. Display for each product the difference between its standard price and the ove
standard price of all products.

```
SELECT PRoductStandardPrice - PriceAvg AS Difference
FROM Product_t, (SELECT AVG(ProductStandardPrice) AS PriceAvg FROM Produ
```

This is not what we typically call a "nested query", but it acts in a similar wa
A couple of notes: MySQL requires that the "derived table" has an alias, in t
Also, the derived table is really just one value, which is "cross-product-ed" with the
This allows the FIELD name PriceAvg to be "tacked on" to each row in Produc
This new temporary table (Product_t cross ProdAvg) is the table against which t

11. Alphabetically, what is the first prodcut name in the Product table?

12. Which orders have been placed since 10/24/2010? (date format: yyyy-mm-dd

13. *Query 13 has been removed.*

14. What furniture does Pine Valley carry that isn't made of cherry?

15. Display all customers for whom we do not know their postal code.

```
SELECT * FROM Customer_T WHERE CustomerPostalCode is NULL;
```

The result is that the empty set, because all customers have a postal code.

```
UPDATE Customer_t
SET    CustomerPostalcode = NULL
WHERE  CustomerId = 1;
```

Or, you could just check the little box in PHPMyAdmin that says Null. :-)

16. List product name, finish, and unit price for all desks and tablers
in the PRODUCT talbe that cost more than $300.

17. Which products in the Product table have a standard price
between $200 and $300?

18a What order numbers are included in the OrderLine table?
Recall that OrderLine has many rows with the same OrderID
since that is part of a composite key.
First, display the duplicates.

18b Note duplicates. So now, get rid of the duplicates.

18c List the unique combinations of order number and order qty in ORderLine:

19. List all customers who live in warmer states. That would be FL, TX, CA, HI)

20a. Same as previous query, but list the customers alphabetically by customer with

20b Same query as above, using column positions instead of names.
(Don't do this unless you have a good reason. It's still a set…)

20c Same query, limit the number or rows in the query response.

20d Same query, skip the first two rows of response, show the next 4.

21 Count the number of customer with addresses in each state to which we ship.

Note: 1 row in the query response corresponds to one group.
Note: If you change one of the states to NULL, it is in its own group.

22. Difference between COUNT(*) and COUNT (*attribute*)

```
SELECT COUNT(*) FROM Customer_t;
SELECT COUNT(CustomerState) FROM Customer_t;
```

Try these when one of the states has a NULL value.

23. Count the number of customers with addresses in each city to which we ship. List the

```
SELECT CustomerState, CustomerCity, COUNT(CustomerCity)
FROM Customer_t
GROUP BY CustomerState, CustomerCity;
```

Change "Santa Clara" to "Sacramento" and rerun the query.

24. Find only states with more than 1 customer.

25. List, in alphabetical order, the product finish and the average standard price for
for selected finishes having an average standard price less than 750.

26. *Often,when specific (and well-defined) views of the database can
be anticipated to be needed on a repeated basis, you can
create a VIEW. Also, for simplicity and for security.
This also really introduces joins.*

What are the data elements necessary to create an invoice for a customer? Sav

```
CREATE VIEW Invoice_V AS
                    SELECT Customer_T.CustomerID, CustomerAddress, Order
                              Product_T.ProductID, ProductStandardPrice, Orde
                    FROM Customer_T, Order_T, OrderLine_T, Product_T
                    WHERE Customer_T.CustomerID=Order_T.CustomerID
                    AND Order_T.OrderID = OrderLine_T.OrderID
                    AND Product_T.ProductID=OrderLine_T.ProductID;
```

Note: Look at the left menu bar under Views. You will see Invoice_v.

27. What are the data elements necessary to create an invoice for order number 1

A view cannot be updatable when:

1 The SELECT clause includes the keyword DISTINCT
2 The SELECT clause contains expressions, including derived columns, aggrega
3 The FROM clause, a subquery of a UNION clause references more than one
4 The FROM clause or a subquery references another view that is not updatea
5 The CREATE VIEW command contains a GROUP BY or a HAVING clause.

Dynamic view: temporarily created each time it's used. Like a query.
Materialized view: data for that view is actually stored.

txt

E clause

: table.

roduct?

:hey?

ate)

products:

----INCORRECT!!!

gate.

rall average

ıct_t) AS ProdAvg;

/.

his case ProdAvg.

Product table.

:t_t.

he query is posed.

)

in state.

cities by state.

each finish

e as a view.

_T.OrderID,
redQuantity

04?

tes, functions, etc.
table
ble.