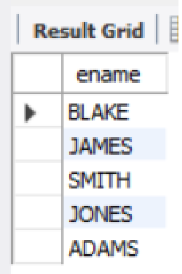
Alan Moss

CSCD 327

Lab 3

1.) Find all the employees in department 30 who do not earn a commission (i.e., comm is null), along with the employees in department 20 whose salaries are less than 3000. (The correct result should have 5 tuples)

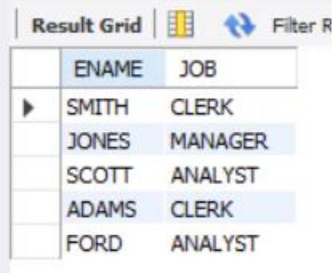
```
SELECT ename FROM EMP WHERE DEPTNO = 30 and COMM IS NULL UNION SELECT ename FROM EMP WHERE DEPTNO = 20 and SAL < 3000;
```



ename
BLAKE
JAMES
SMITH
JONES
ADAMS

2.) List the ENAME and JOB of employees assigned to department number 20. (The correct result should have 5 tuples)

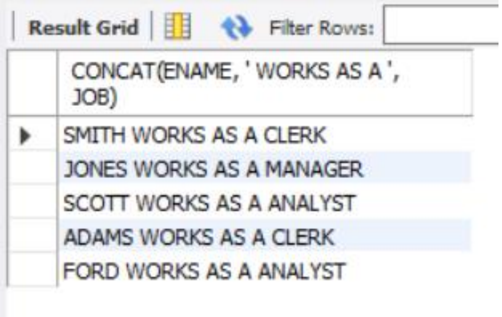
```
SELECT ENAME, JOB FROM EMP WHERE DEPTNO = 20;
```



ENAME	JOB
SMITH	CLERK
JONES	MANAGER
SCOTT	ANALYST
ADAMS	CLERK
FORD	ANALYST

3.) Can you display the query result from Question 2 as the following? (Hint: MySQL supports a function called CONCAT to concatenate values from multiple columns.)

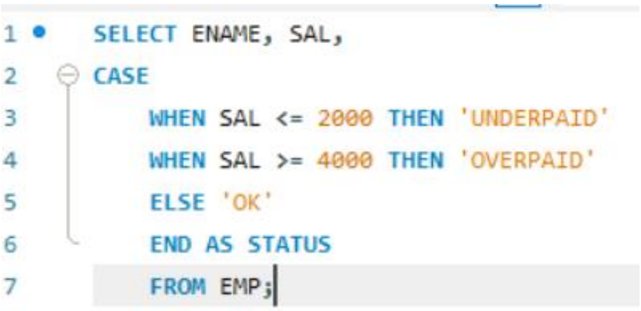
```
SELECT CONCAT(ENAME, ' WORKS AS A ', JOB) FROM EMP WHERE DEPTNO = 20;
```



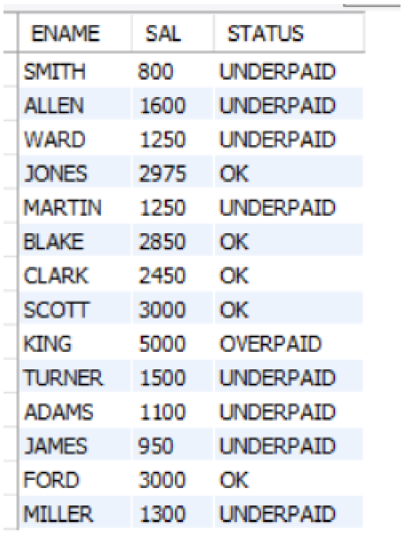
CONCAT(ENAME, ' WORKS AS A ', JOB)
SMITH WORKS AS A CLERK
JONES WORKS AS A MANAGER
SCOTT WORKS AS A ANALYST
ADAMS WORKS AS A CLERK
FORD WORKS AS A ANALYST

4.) Sometimes you want to perform IF-ELSE operations on values in your SELECT statement. For example, you would like to produce a result set such that, if an employee is paid $2000 or less, a message of "UNDERPAID" is returned, if an employee is paid $4000 or more, a message of "OVERPAID" is returned, if they make somewhere in between, then "OK" is returned. The result set should look like this:

Hint: Use the CASE expression to perform conditional logic directly in the SELECT statement. CASE is combined with WHEN and THEN to specify the condition.

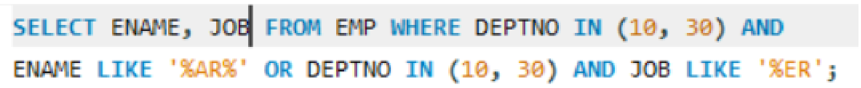


```
SELECT ENAME, SAL,
CASE
    WHEN SAL <= 2000 THEN 'UNDERPAID'
    WHEN SAL >= 4000 THEN 'OVERPAID'
    ELSE 'OK'
    END AS STATUS
    FROM EMP;
```

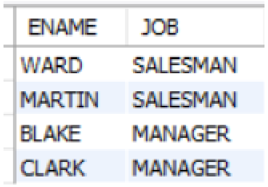


ENAME	SAL	STATUS
SMITH	800	UNDERPAID
ALLEN	1600	UNDERPAID
WARD	1250	UNDERPAID
JONES	2975	OK
MARTIN	1250	UNDERPAID
BLAKE	2850	OK
CLARK	2450	OK
SCOTT	3000	OK
KING	5000	OVERPAID
TURNER	1500	UNDERPAID
ADAMS	1100	UNDERPAID
JAMES	950	UNDERPAID
FORD	3000	OK
MILLER	1300	UNDERPAID

5.) Find the ENAME and JOB of all the employees in departments 10 and 30 and return only those who have either an "AR" somewhere in their name or a job title ending with "ER". (The correct result should have 4 tuples.)



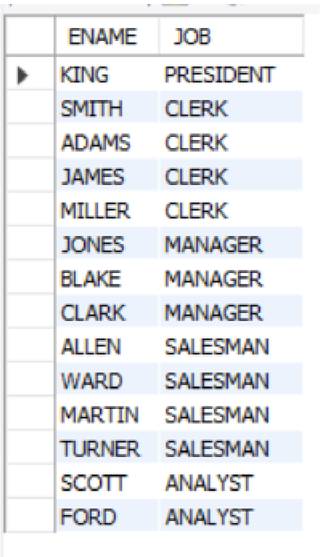
```
SELECT ENAME, JOB FROM EMP WHERE DEPTNO IN (10, 30) AND
ENAME LIKE '%AR%' OR DEPTNO IN (10, 30) AND JOB LIKE '%ER';
```



ENAME	JOB
WARD	SALESMAN
MARTIN	SALESMAN
BLAKE	MANAGER
CLARK	MANAGER

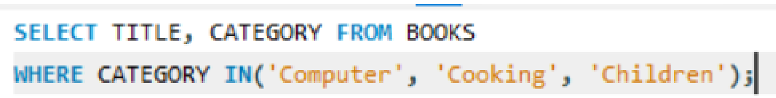
6.) Return employee names and jobs from table EMP and sort by the last THREE characters in the job field. The result set should look like the following:
Hint: MySQL supports SUBSTR function and LENGTH function. SUBSTR(str, pos): Select all characters from<str> starting with position <pos>.LENGTH(str):Return the length of <str>

```
SELECT ENAME, JOB FROM EMP ORDER BY SUBSTR(JOB, -3, 3);
```

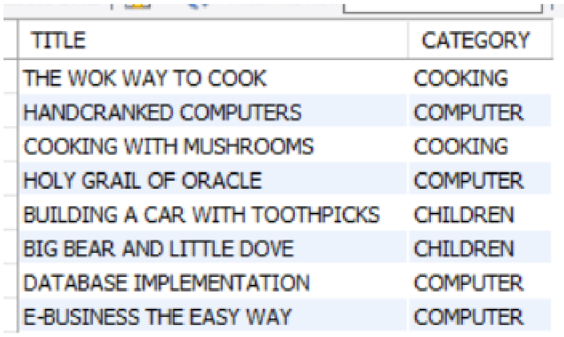


	ENAME	JOB
▶	KING	PRESIDENT
	SMITH	CLERK
	ADAMS	CLERK
	JAMES	CLERK
	MILLER	CLERK
	JONES	MANAGER
	BLAKE	MANAGER
	CLARK	MANAGER
	ALLEN	SALESMAN
	WARD	SALESMAN
	MARTIN	SALESMAN
	TURNER	SALESMAN
	SCOTT	ANALYST
	FORD	ANALYST

7.) Find all the books that are in the Computer, Cooking, or Children categories. List each book's title and category. (The correct result should have 8 tuples.)

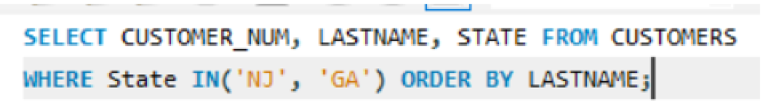


```
SELECT TITLE, CATEGORY FROM BOOKS
WHERE CATEGORY IN('Computer', 'Cooking', 'Children');
```

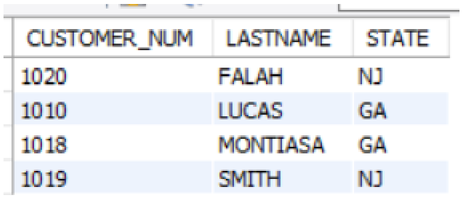


TITLE	CATEGORY
THE WOK WAY TO COOK	COOKING
HANDCRANKED COMPUTERS	COMPUTER
COOKING WITH MUSHROOMS	COOKING
HOLY GRAIL OF ORACLE	COMPUTER
BUILDING A CAR WITH TOOTHPICKS	CHILDREN
BIG BEAR AND LITTLE DOVE	CHILDREN
DATABASE IMPLEMENTATION	COMPUTER
E-BUSINESS THE EASY WAY	COMPUTER

8.) Find all the customers who live in Georgia or New Jersey. Put the results in ascending order by last name. List each customer's customer number, last name, and state. (The correct result should have 4 tuples.)

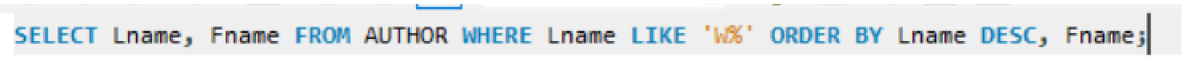


```
SELECT CUSTOMER_NUM, LASTNAME, STATE FROM CUSTOMERS
WHERE State IN('NJ', 'GA') ORDER BY LASTNAME;
```

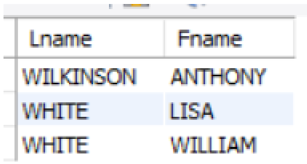


CUSTOMER_NUM	LASTNAME	STATE
1020	FALAH	NJ
1010	LUCAS	GA
1018	MONTIASA	GA
1019	SMITH	NJ

9.) Find all authors whose last name begins with letter "W". Put the results in descending order of last name, then ascending order of first name. List each author's last name and first name. (The correct result should have 3 tuples.)

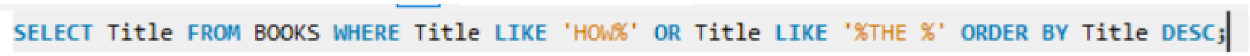


```
SELECT Lname, Fname FROM AUTHOR WHERE Lname LIKE 'W%' ORDER BY Lname DESC, Fname;
```

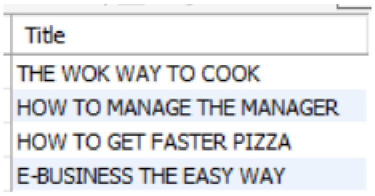


Lname	Fname
WILKINSON	ANTHONY
WHITE	LISA
WHITE	WILLIAM

10.) Use a search pattern to find book titles begin with the word "HOW" or include the word "THE". Sort the list by title in descending order. (The correct result should have 4 tuples.)



```
SELECT Title FROM BOOKS WHERE Title LIKE 'HOW%' OR Title LIKE '%THE %' ORDER BY Title DESC;
```

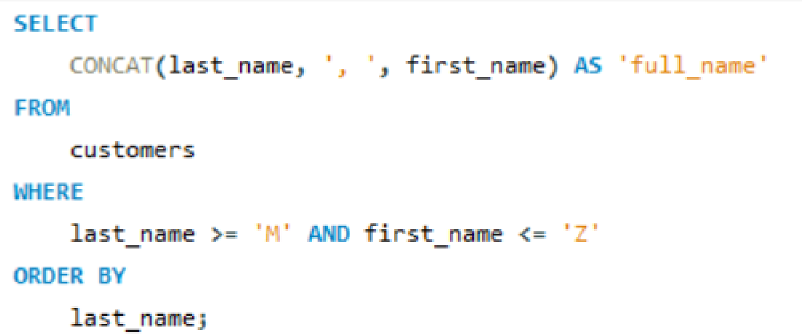


Title
THE WOK WAY TO COOK
HOW TO MANAGE THE MANAGER
HOW TO GET FASTER PIZZA
E-BUSINESS THE EASY WAY

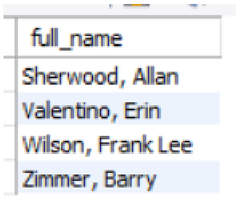
11.) Return one column from the Customers table named full_name that joins the last_name and first_name columns. Format this column with the last name, a comma, a space, and the first name like this:

Doe, John

Sort the result set by last name in ascending order and return only the customers whose last name begins with letters from M to Z. (The correct result should have 4 tuples.)



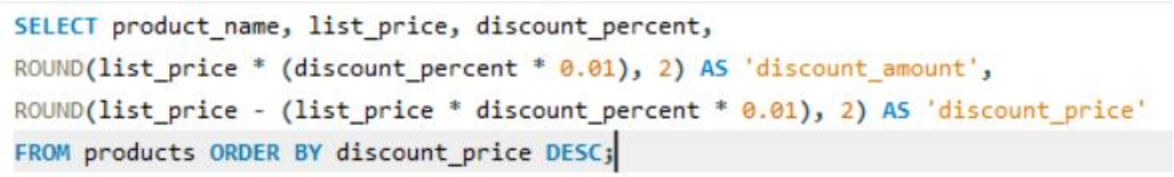
```
SELECT
    CONCAT(last_name, ', ', first_name) AS 'full_name'
FROM
    customers
WHERE
    last_name >= 'M' AND first_name <= 'Z'
ORDER BY
    last_name;
```



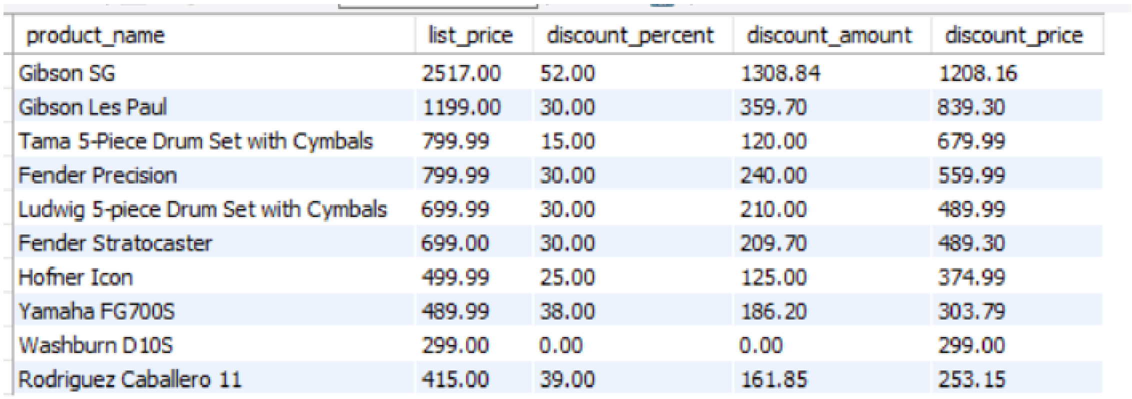
full_name
Sherwood, Allan
Valentino, Erin
Wilson, Frank Lee
Zimmer, Barry

12.) Return these column names and data from the Products table:

Round the discount_amount and price_after_discount columns to 2 decimal places. Sort the result set by price_after_discount in descending order. (The correct result is given as a reference.)

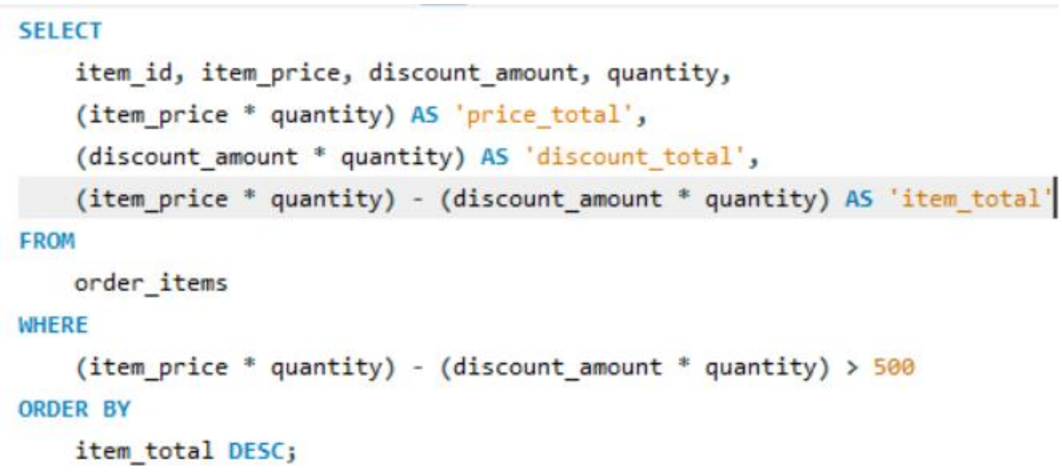


```
SELECT product_name, list_price, discount_percent,
ROUND(list_price * (discount_percent * 0.01), 2) AS 'discount_amount',
ROUND(list_price - (list_price * discount_percent * 0.01), 2) AS 'discount_price'
FROM products ORDER BY discount_price DESC;
```

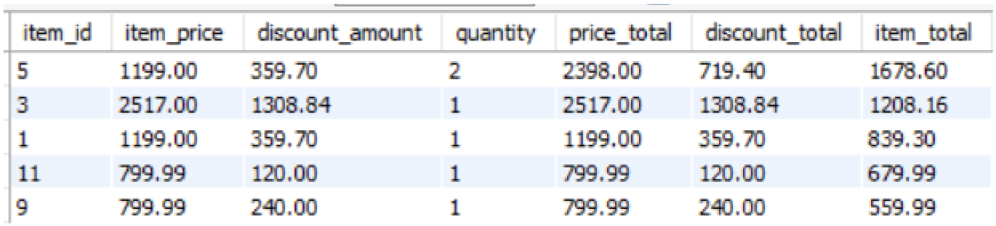


product_name	list_price	discount_percent	discount_amount	discount_price
Gibson SG	2517.00	52.00	1308.84	1208.16
Gibson Les Paul	1199.00	30.00	359.70	839.30
Tama 5-Piece Drum Set with Cymbals	799.99	15.00	120.00	679.99
Fender Precision	799.99	30.00	240.00	559.99
Ludwig 5-piece Drum Set with Cymbals	699.99	30.00	210.00	489.99
Fender Stratocaster	699.00	30.00	209.70	489.30
Hofner Icon	499.99	25.00	125.00	374.99
Yamaha FG700S	489.99	38.00	186.20	303.79
Washburn D10S	299.00	0.00	0.00	299.00
Rodriguez Caballero 11	415.00	39.00	161.85	253.15

13.)Write a SELECT statement that returns these column names and data from the Order_Items table:
Only return rows where the item_total is greater than 500.Sort the result by item_total in descending order. (The correct result is given as a reference.)



```
SELECT
    item_id, item_price, discount_amount, quantity,
    (item_price * quantity) AS 'price_total',
    (discount_amount * quantity) AS 'discount_total',
    (item_price * quantity) - (discount_amount * quantity) AS 'item_total'
FROM
    order_items
WHERE
    (item_price * quantity) - (discount_amount * quantity) > 500
ORDER BY
    item_total DESC;
```



item_id	item_price	discount_amount	quantity	price_total	discount_total	item_total
5	1199.00	359.70	2	2398.00	719.40	1678.60
3	2517.00	1308.84	1	2517.00	1308.84	1208.16
1	1199.00	359.70	1	1199.00	359.70	839.30
11	799.99	120.00	1	799.99	120.00	679.99
9	799.99	240.00	1	799.99	240.00	559.99