

CSCD 327 Lab #3 (13 points)

Include the SQL statements AND the query results in your submission.

Section 1: Use database *YourUsername_1* to complete the following queries in SQL.

1. Find all the employees in department 10, along with any employees who earn a commission (i.e., comm isn't null), along with any employees in department 20 whose salary are at most \$2000.
2. List the ENAME and JOB of employees assigned to department number 10.
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.)

Works_As
CLARK WORKS AS A MANAGER
KING WORKS AS A PRESIDENT
MILLER WORKS AS A CLERK

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:

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

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.

- Find all the employees in departments 10 and 20, and return only those that have either an "I" somewhere in their name or a job title ending with "ER".
- 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:

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

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>.

Section 2: Use database *YourUsername_3* to complete the following queries in SQL.

- Find all the books that are **NOT** in the Fitness category. List each book title and category.
- 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.
- List all authors whose last name contains the letter pattern "IN". Put the results in order of last name, then first name. List each author's last name and first name.

10. Use a search pattern to find any book title with "A" for the second letter and "N" for the fourth letter. List each book's ISBN and title. Sort the list by title in descending order.

Section 3: Use database *YourUsername_5* to complete the following queries in SQL.

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.

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

product_name	The product_name column
list_price	The list_price column
discount_percent	The discount_percent column
discount_amount	A column that's calculated from the previous two columns
discount_price	A column that's calculated from the previous three columns

Round the discount_amount and discount_price columns to 2 decimal places. Sort the result set by discount price in descending order.

13. Write a SELECT statement that returns these column names and data from the Order_Items table:

item_id	The item_id column
item_price	The item_price column
discount_amount	The discount_amount column
quantity	The quantity column
price_total	A column that's calculated by multiplying the item price by the quantity
discount_total	A column that's calculated by multiplying the discount amount by the quantity
item_total	A column that's calculated by subtracting the discount amount from the item price and then multiplying by the quantity

Only return rows where the item_total is greater than 500. Sort the result set by item total in descending order.