CSCD 327 Lab #3 (13 points)

Due: 11:59pm on October 21, 2020 (Submit Online)

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

Section 1: Use database *employeeDB* 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.

empno	ename	job	mgr	hiredate	sal	comm	deptno
7369	SMITH	CLERK	7902	1980-12-17	800	*NULL*	20
7499	ALLEN	SALESMAN	7698	1981-02-20	1600	300	30
7521	WARD	SALESMAN	7698	1981-02-22	1250	500	30
7654	MARTIN	SALESMAN	7698	1981-09-28	1250	1400	30
7782	CLARK	MANAGER	7839	1981-06-09	2450	*NULL*	10
7839	KING	PRESIDENT	*NULL*	1981-11-17	5000	*NULL*	10
7844	TURNER	SALESMAN	7698	1981-09-08	1500	0	30
7876	ADAMS	CLERK	7788	1983-01-12	1100	*NULL*	20
7934	MILLER	CLERK	7782	1982-01-23	1300	*NULL*	10

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.

5. 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".

ename	job
SMITH	CLERK
JONES	MANAGER
CLARK	MANAGER
KING	PRESIDENT
MILLER	CLERK

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:

```
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 *booksDB* to complete the following queries in SQL.

7. Find all the books that are **NOT** in the Fitness category. List each book title and category.

title	category
HOW TO GET FASTER PIZZA	SELF HELP
THE WOK WAY TO COOK	COOKING
REVENGE OF MICKEY	FAMILY LIFE
HANDCRANKED COMPUTERS	COMPUTER
SHORTEST POEMS	LITERATURE
PAINLESS CHILD-REARING	FAMILY LIFE
COOKING WITH MUSHROOMS	COOKING
HOLY GRAIL OF ORACLE	COMPUTER
BUILDING A CAR WITH TOOTHPICKS	CHILDREN
BIG BEAR AND LITTLE DOVE	CHILDREN
DATABASE IMPLEMENTATION	COMPUTER
HOW TO MANAGE THE MANAGER	BUSINESS
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.

customer_num	lastname	state
1020	FALAH	NJ
1010	LUCAS	GA
1018	MONTIASA	GA
1019	SMITH	NJ

9. 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.

lname	fname
AUSTIN	JAMES
MARTINEZ	SHEILA
ROBINSON	ROBERT
WILKINSON	ANTHONY

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.

isbn	title
2491748320	PAINLESS CHILD-REARING

Section 3: Use database *productsDB* 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.

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

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.

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:

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.

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