Julian Welge

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.

```
80 •  select EMPNO, ENAME from EMP where DEPTNO = 10 or COMM != NULL or COMM > 0 or (DEPTNO = 20 and SAL <= 2000);
81
```

100% 16:80

Result Grid Filter Rows: Search Export:

EMPNO	ENAME
7369	SMITH
7499	ALLEN
7521	WARD
7654	MARTIN
7782	CLARK
7839	KING
7876	ADAMS
7934	MILLER

2. ListtheENAMEandJOBofemployeesassignedtodepartmentnumber 10.

```
81 •  select ENAME, JOB from EMP where DEPTNO = 10;
82
```

100% 1:81

Result Grid Filter Rows: Search Export:

ENAME	JOB
CLARK	MANAGER
KING	PRESIDENT
MILLER	CLERK

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

```
select CONCAT(ENAME, " works as a ", JOB) as Works_As from EMP where DEPTNO = 10;
```

100% 1:81

Result Grid Filter Rows: Search Export:

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

```
select ENAME, SAL,
CASE
    when SAL <= 2000 then "UNDERPAID"
    when SAL >= 4000 then "OVERPAID"
    else "OK"
end as STATUS_
from EMP;
```

100% 1:82

Result Grid Filter Rows: Search Export:

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

"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:
3000 OK

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".

```
89 •  select ENAME, JOB from EMP
90    where DEPTNO in (10, 20) and (ENAME like "%I%" or JOB like "%er");
91
```

100% 1:89

Result Grid Filter Rows: Search Export:

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 ---------- KING SMITH ADAMS JAMES MILLER JONES CLARK BLAKE ALLEN MARTIN WARD TURNER SCOTT FORD

JOB --------- PRESIDENT CLERK CLERK CLERK CLERK MANAGER MANAGER MANAGER SALESMAN SALESMAN SALESMAN SALESMAN ANALYST ANALYST

Hint: MySQL supports SUBSTR function and LENGTH function.

```
91 •  select ENAME, JOB from EMP
92    ORDER BY SUBSTR(JOB, (LENGTH(JOB)-2));
```

100% 1:91

Result Grid Filter Rows: Search Export:

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

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.

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

```
select title, category from BOOKS
where category != "FITNESS";
```

00% 29:3

Result Grid Filter Rows: Search E

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.

```
select customer_num, Lastname, firstname, state from CUSTOMERS
where state in ("GA", "NJ")
```

100% 1:4

Result Grid Filter Rows: Search Export:

customer_num	Lastname	firstname	state
1020	FALAH	KENNETH	NJ
1010	LUCAS	JAKE	GA
1018	MONTIASA	GREG	GA
1019	SMITH	JENNIFER	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.

```
7 •  select lname, fname from AUTHOR
8    where lname like "%IN%"
9    ORDER BY lname;
```

100% 1:7

Result Grid Filter Rows: Search

lname	fname
AUSTIN	JAMES
MARTINEZ	SHEILA
ROBINSON	ROBERT
WILKINSON	ANTHONY

2

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.

```
10 •  select title, ISBN from BOOKS
11    where title like "_A_N%";
```

100% 1:10

Result Grid Filter Rows: Search

title	ISBN
PAINLESS CHILD-REARING	2491748320

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.

```
select CONCAT(last_name, ", ", first_name) as full_name from customers
where last_name between "M%" and "Z%"
order by last_name;
```

100% 1:2

Result Grid Filter Rows: Search Export:

full_name
Sherwood, Allan
Valentino, Erin
Wilson, Frank Lee

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

product_name list_price discount_percent discount_amount

discount_price

The product_name column

The list_price column

The discount_percent column

A column that’s calculated from the previous two columns

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.

```
select product_name, list_price, discount_percent,
Round(concat(list_price * (0.01 * discount_percent)), 2) as discount_amount,
Round(concat(list_price - (list_price * (0.01 * discount_percent))), 2) as discount_price
from products;
```

100% 1:5

Result Grid Filter Rows: Search Export:

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

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

item_id item_price discount_amount quantity price_total

discount_total item_total

The item_id column

The item_price column

The discount_amount column

The quantity column

A column that’s calculated by multiplying the item price by the quantity

A column that’s calculated by multiplying the discount amount by the quantity

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.

```sql
select item_id, item_price, discount_amount, quantity,
concat(item_price * quantity) as price_total, concat(discount_amount * quantity) as discount_total,
concat((item_price - discount_amount) * quantity) as item_total
from order_items
where ((item_price - discount_amount) * quantity) > 500
order by item_total DESC;
```

100% 1:9

Result Grid Filter Rows: Search Export:

item_id	item_price	discount_amou...	quantity	price_total	discount_to...	item_total
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
5	1199.00	359.70	2	2398.00	719.40	1678.60
3	2517.00	1308.84	1	2517.00	1308.84	1208.16

3