Lab # 04:

Lab Title:

“Relationships, Simple Query Designing and Advance Queries“

Lab Objectives:

The objectives of this lab are as follows

- Learn about the different types of relationships
- Retrieving data based on some criteria
- Performing some calculations on the attributes of tables

Introduction:

Relationships refer to how different pieces of information are connected or linked to each other. Just like friendships between people, data in a database can have relationships. These relationships define how data from one 'thing' is associated or linked to data from another.

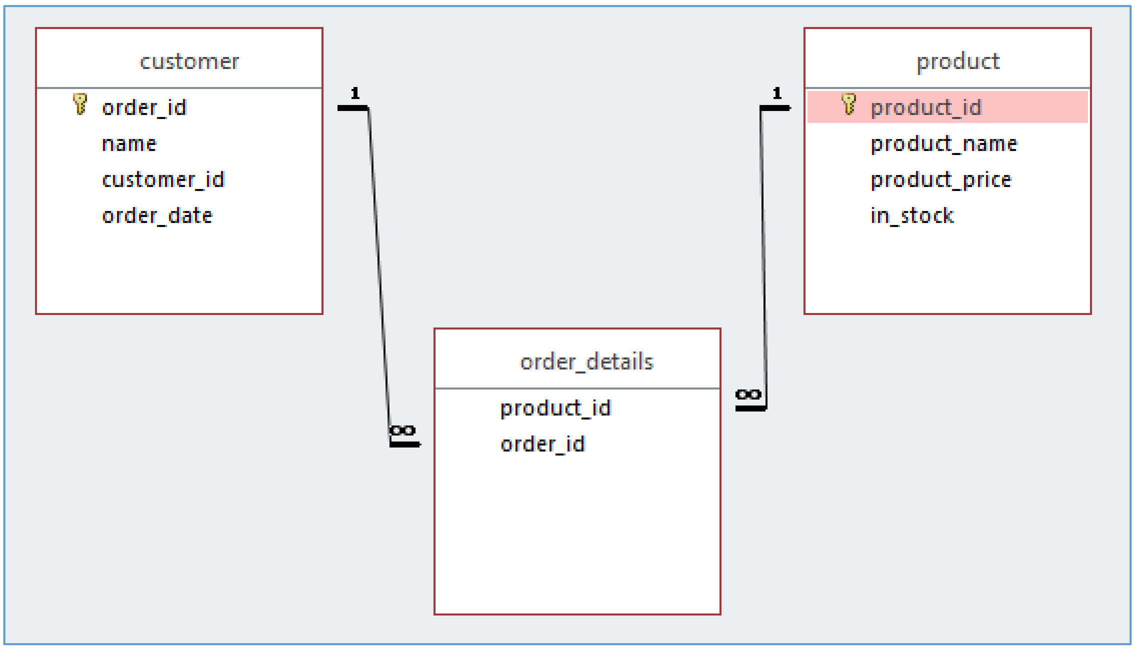
Query designing is like crafting specific questions to get the exact information you need from a database. Imagine you're asking a librarian for books about space. You'd ask a precise question so they can find the right books for you.

Advanced queries in databases are like solving complex puzzles. Once you're comfortable with basic queries (like finding specific students in a class), advanced queries take it up a notch. It's like trying to find all the students who got an A in both Math and Science, or sorting products not just by price but also by popularity. These queries involve more intricate commands and logic. You might combine different conditions, use calculations, or even gather information from multiple tables at once. They help uncover deeper insights from the data, allowing you to do more complex analysis and get very specific results.

In-Lab Tasks:

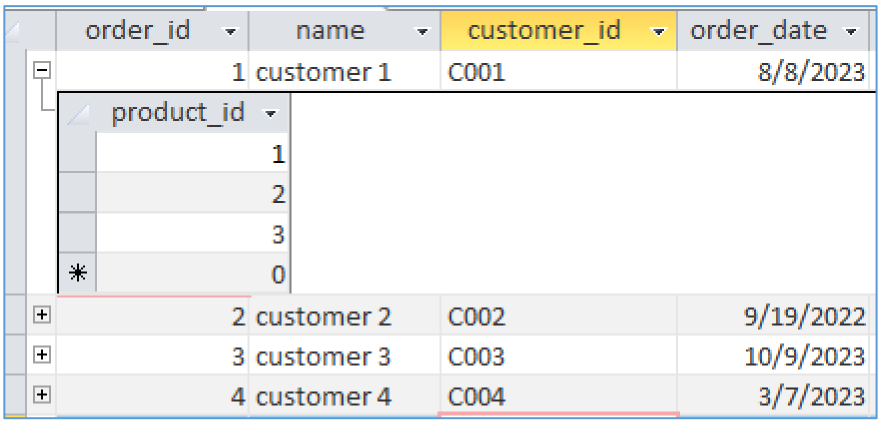
Task#01:

Relationship:



Tables:

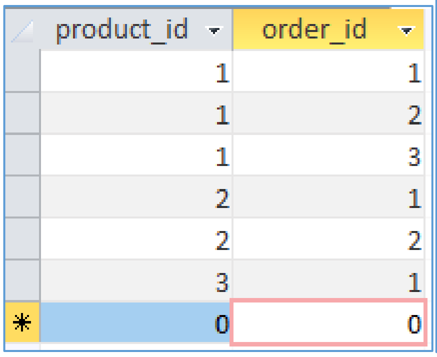
Customer Table:



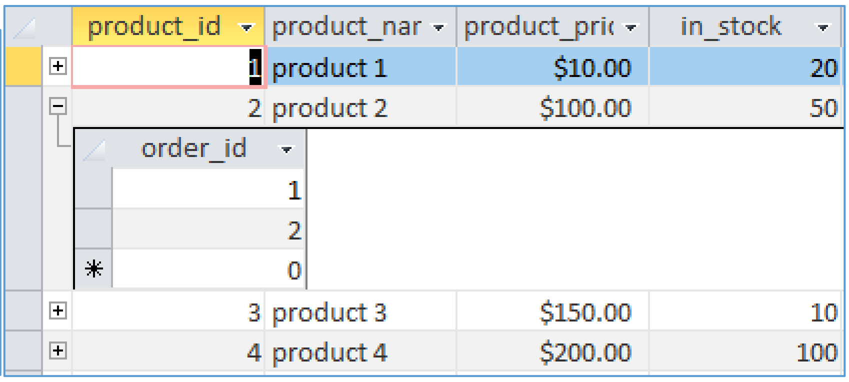
order_id	name	customer_id	order_date
1	customer 1	C001	8/8/2023
2	customer 2	C002	9/19/2022
3	customer 3	C003	10/9/2023
4	customer 4	C004	3/7/2023

product_id
1
2
3
✱ 0

Junction Table: **Product Table:**



product_id	order_id
1	1
1	2
1	3
2	1
2	2
3	1
✱ 0	0

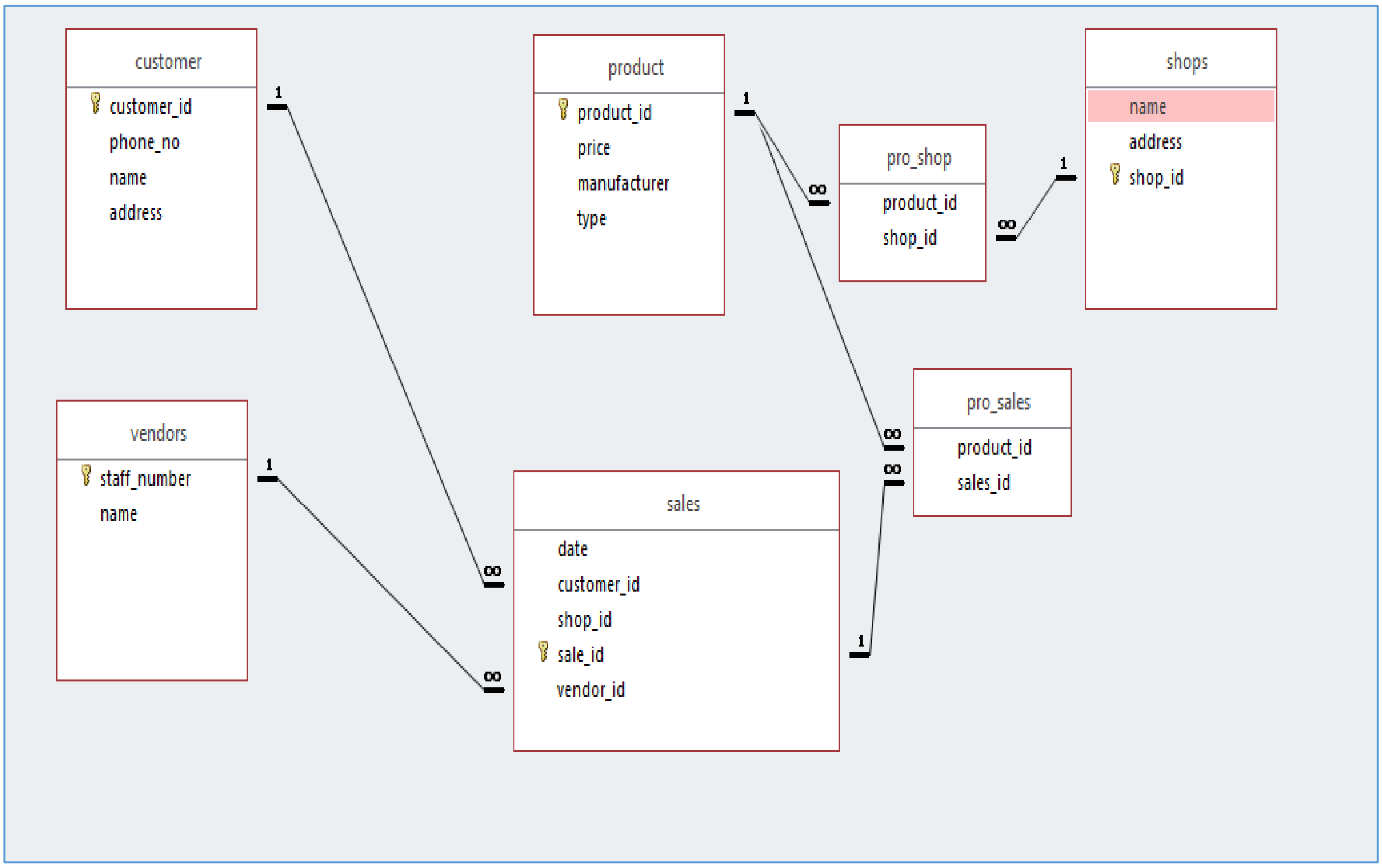


product_id	product_nar	product_pri	in_stock
1	product 1	$10.00	20
2	product 2	$100.00	50
3	product 3	$150.00	10
4	product 4	$200.00	100

order_id
1
2
✱ 0

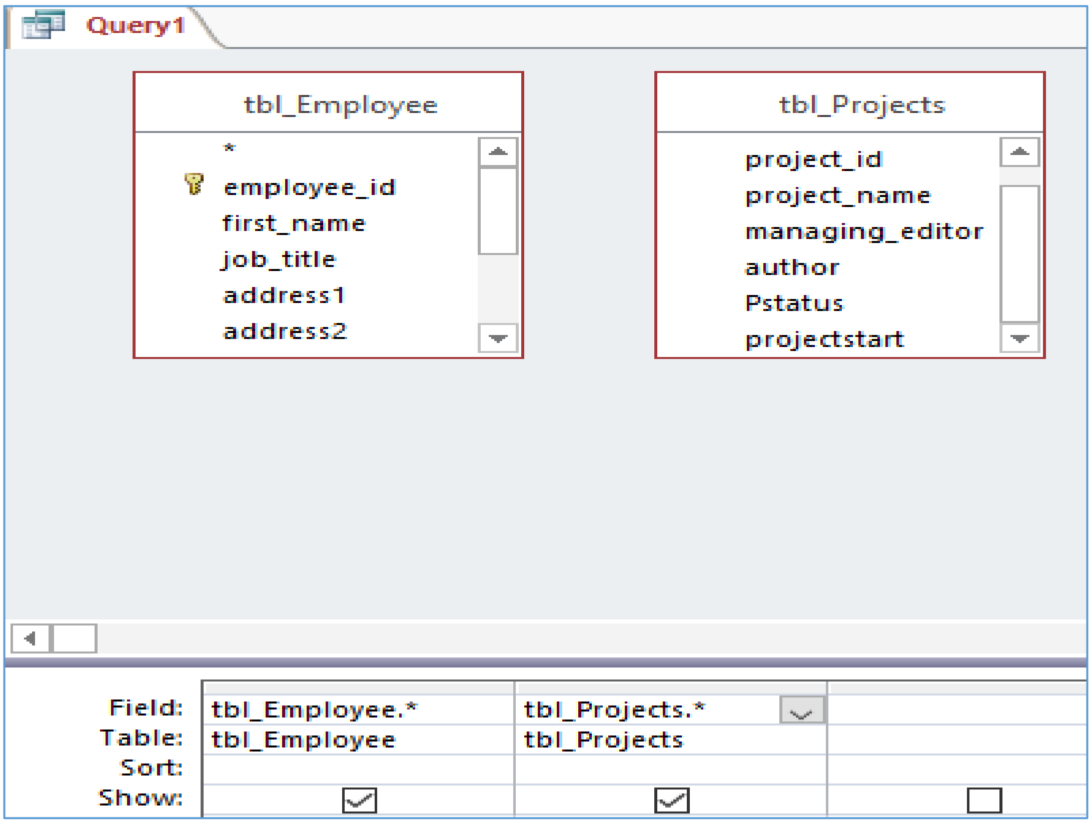
Task#02:

Relationships:



Task#03:

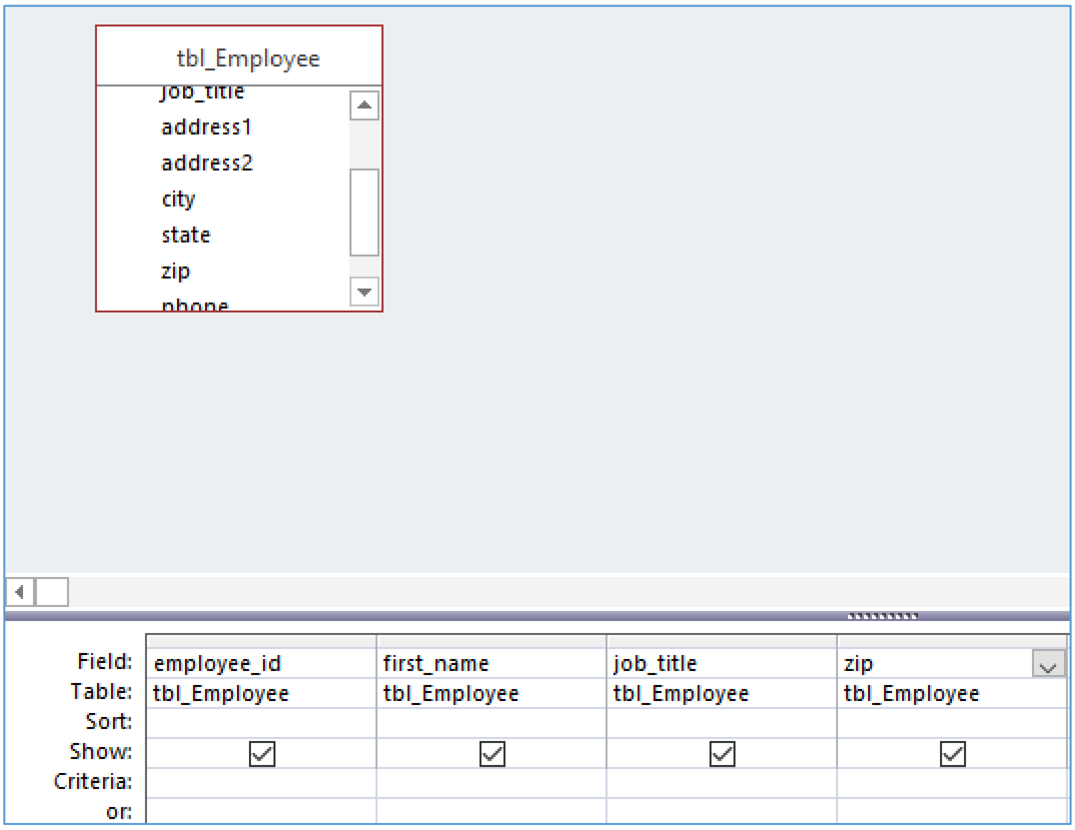
We will be sowing all the data of table employee and projects, the following query will be used



Field:	tbl_Employee.*	tbl_Projects.*	
Table:	tbl_Employee	tbl_Projects	
Sort:			
Show:	✓	✓	☐

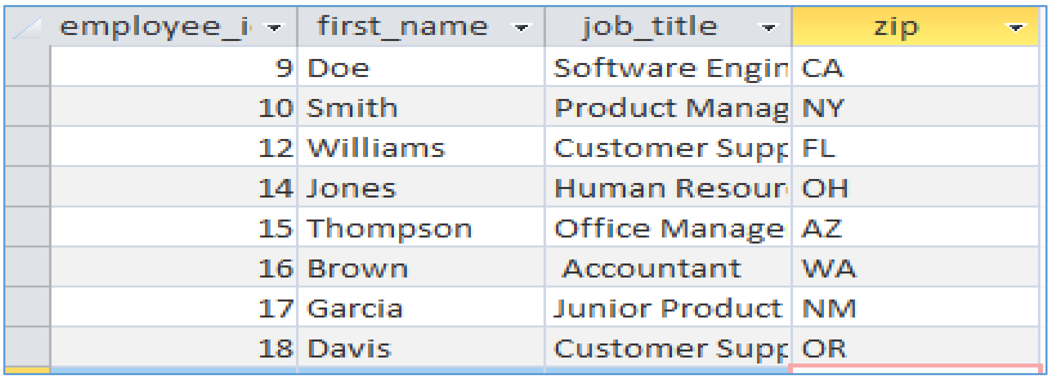
Task#04:

We will be showing the employee ID, employee name, job title and zip code, to do this the following query will be used



Field:	employee_id	first_name	job_title	zip
Table:	tbl_Employee	tbl_Employee	tbl_Employee	tbl_Employee
Sort:				
Show:	✓	✓	✓	✓
Criteria:				
or:				

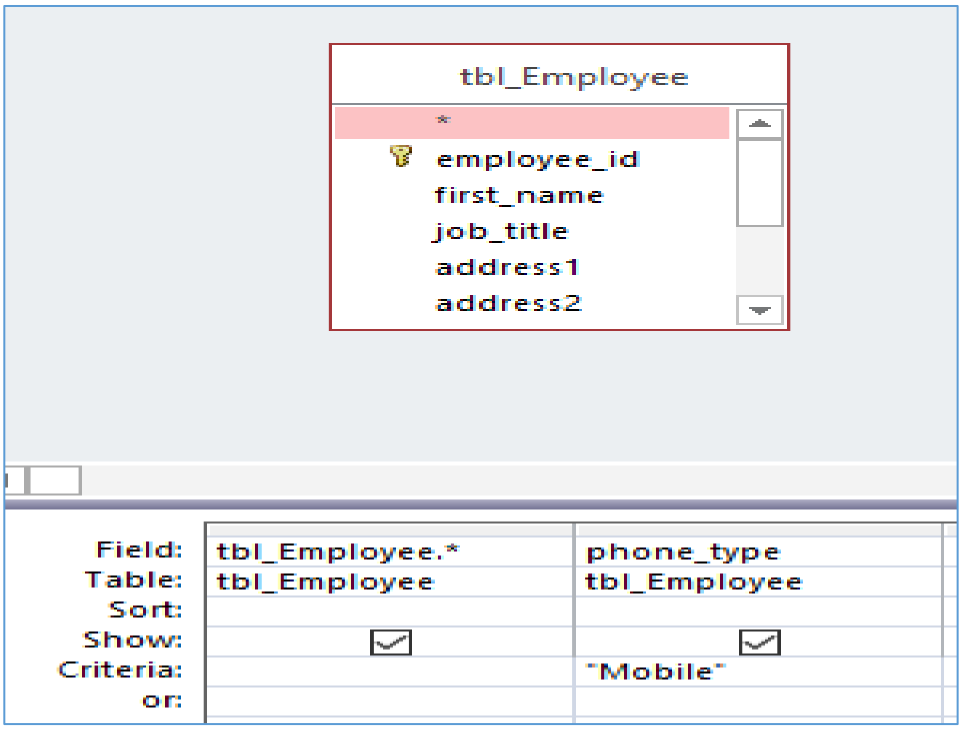
Output:



employee_i	first_name	job_title	zip
9	Doe	Software Engir	CA
10	Smith	Product Manag	NY
12	Williams	Customer Supp	FL
14	Jones	Human Resour	OH
15	Thompson	Office Manage	AZ
16	Brown	Accountant	WA
17	Garcia	Junior Product	NM
18	Davis	Customer Supp	OR

Task#05:

If any employee has phone type as mobile we will show them only, the following query will be used



tbl_Employee

*

employee_id

first_name

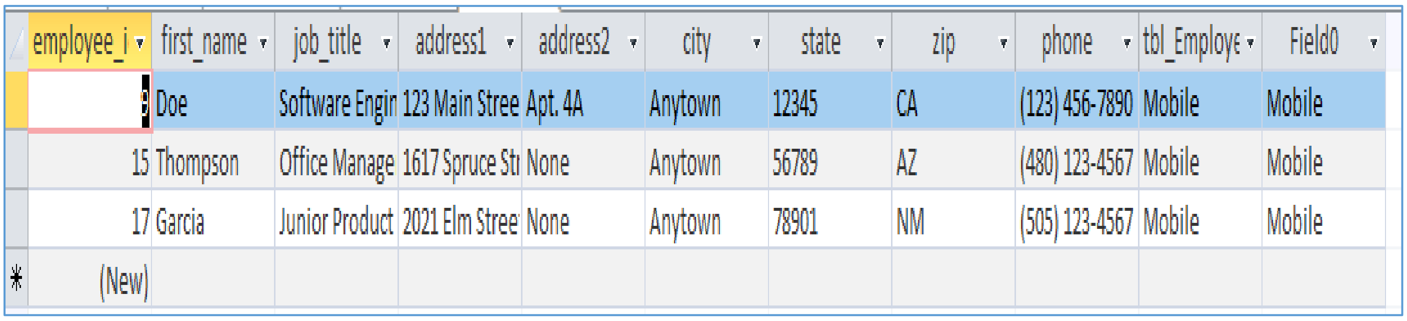
job_title

address1

address2

Field:	tbl_Employee.*	phone_type
Table:	tbl_Employee	tbl_Employee
Sort:		
Show:	✓	✓
Criteria:		"Mobile"
or:		

Output:

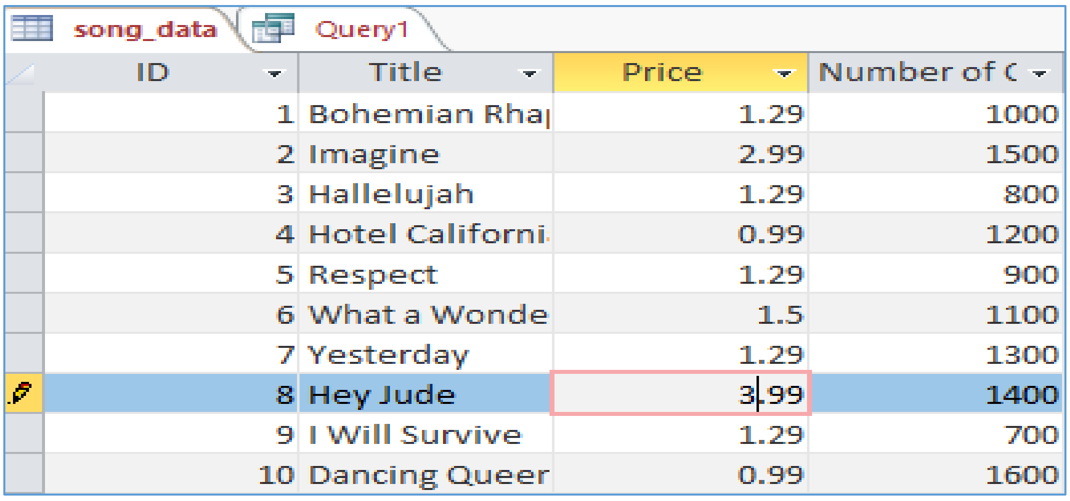


employee_i	first_name	job_title	address1	address2	city	state	zip	phone	tbl_Employe	Field0
9	Doe	Software Engir	123 Main Stree	Apt. 4A	Anytown	12345	CA	(123) 456-7890	Mobile	Mobile
15	Thompson	Office Manage	1617 Spruce St	None	Anytown	56789	AZ	(480) 123-4567	Mobile	Mobile
17	Garcia	Junior Product	2021 Elm Stree	None	Anytown	78901	NM	(505) 123-4567	Mobile	Mobile
(New)										

Post-Lab Tasks:

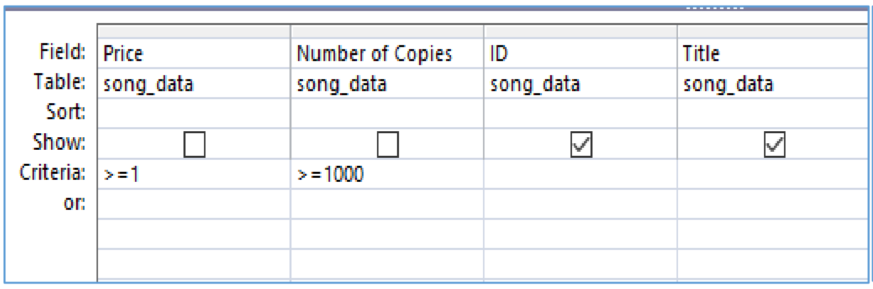
Task#01:

In the task we will have to print the songs which satisfy the following condition , price >= 1.00 and copies >= 1000.the output table will have only the title and id .


song_data | Query1

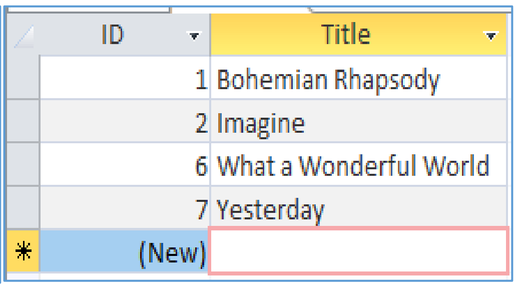
ID	Title	Price	Number of C
1	Bohemian Rha	1.29	1000
2	Imagine	2.99	1500
3	Hallelujah	1.29	800
4	Hotel Californi	0.99	1200
5	Respect	1.29	900
6	What a Wonde	1.5	1100
7	Yesterday	1.29	1300
8	Hey Jude	3.99	1400
9	I Will Survive	1.29	700
10	Dancing Queer	0.99	1600

Query Design



Field:	Price	Number of Copies	ID	Title
Table:	song_data	song_data	song_data	song_data
Sort:				
Show:	☐	☐	☑	☑
Criteria:	>=1	>=1000		
or:				

Output

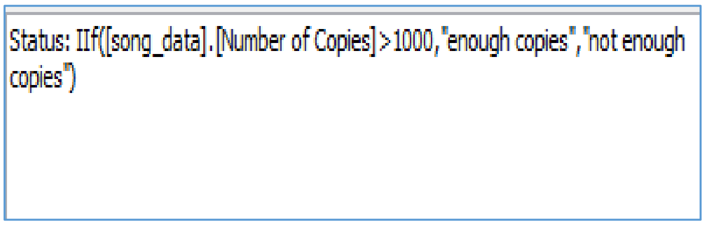


ID	Title
1	Bohemian Rhapsody
2	Imagine
6	What a Wonderful World
7	Yesterday
(New)	

Task # 02:

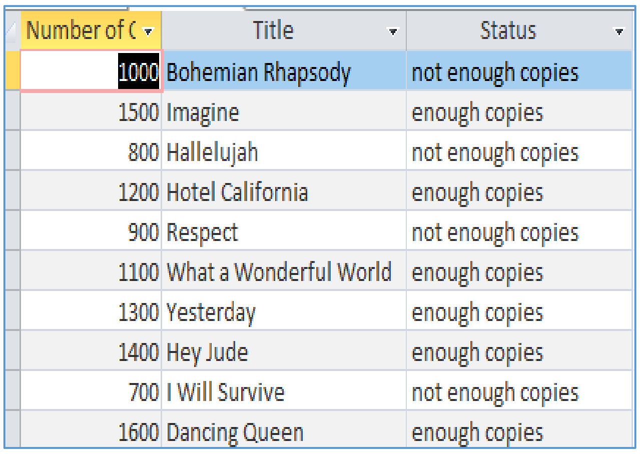
If the number of copies are greater than 1000 than “enough copies” else “not enough copies”, also modifying the column name , the new column name is “Status”.

Expression builder:



```
Status: IIf([song_data].[Number of Copies]>1000,"enough copies","not enough
copies")
```

Output:



Number of C	Title	Status
1000	Bohemian Rhapsody	not enough copies
1500	Imagine	enough copies
800	Hallelujah	not enough copies
1200	Hotel California	enough copies
900	Respect	not enough copies
1100	What a Wonderful World	enough copies
1300	Yesterday	enough copies
1400	Hey Jude	enough copies
700	I Will Survive	not enough copies
1600	Dancing Queen	enough copies