

## Question 1:

Imagine you are an event planner at a company, and you need to organize a special event for employees working in a Sales department. You want to retrieve the names of all employees in this particular department to send them event invitations or plan department-specific activities.

### Requirement :

- WHERE Clause with Column Name
- WHERE Clause with Literal Value

### Output Format :

- It should display the first name, as well as last name who belongs to sales department.

### Table Structure:

- EmployeeID, FirstName, LastName, Department, Salary

### Input Table :

EmployeeID	FirstName	LastName	Department	Salary
1	Alice	Smith	HR	60000
2	Bob	Johnson	Engineering	75000
3	Charlie	Brown	HR	62000
4	David	Davis	Sales	58000
5	Eve	White	Engineering	72000

**Title for Question 1:** Query Equalization ( WHERE with Column Name and WHERE with Literal Value)

### Solution:

```
SELECT FirstName, LastName
FROM Employees
WHERE Department = 'Sales';
SELECT FirstName, LastName
FROM Employees
WHERE 'Sales' = Department;
```

### TestCases:

S.No	Inputs	Outputs
1		FirstName   LastName -----   ----- David   Davis FirstName   LastName -----   ----- David   Davis

S.No	Inputs	Outputs
2		
3		
4		
5		
6		

**White List:**

**Black List:**

**Question 2:**

Imagine you are an HR manager at a company, and you want to determine the lowest salary among employees in the HR department. You need this information for budget planning and salary bench marking purposes.

**Requirements:?????**

- Using Where class
- Using Having class

**Output Format :**

The result of this query will provide you with the lowest salary within the HR department.

**Table Structure:**

EmployeeID, Name, Department, Salary

**Input Table:**

EmployeeID	Name	Department	Salary
1	Alice	HR	65,000.00
2	Bob	Engineering	75,000.00
3	Charlie	HR	62,000.00
4	David	Sales	58,000.00
5	Eve	Engineering	72,000.00

**Title for Question 2:** Query Equalization ( where and having )

**Solution:**

```
SELECT MIN(Salary) AS MinSalary
FROM Employees
WHERE Department = 'HR' ;
```

### TestCases:

S.No	Inputs	Outputs
1		MinSalary ----- 62000 MinSalary ----- 62000
2		
3		
4		
5		
6		

### White List:

### Black List:

---

### Question 3:

Suppose you are the administrator of a university's enrollment system. You want to identify students who are eligible for enrollment based on their age. In this scenario, you're retrieving the first and last names of students who are older than 18 years, indicating they meet the age requirement for enrollment in university courses. This query helps you generate a list of eligible students for the upcoming semester.

### Requirements:

- Straightforward condition
- Negation condition

### Output Format :

It should display the first name, as well as last name where age > 18.

### Table Structure:

StudentID, FirstName, LastName, Age

### Input Table:

ID	First Name	Last Name	Age
1	Alice	Johnson	16
2	Bob	Smith	17
3	Charlie	Brown	18
4	David	Wilson	19
5	Eve	Taylor	17

**Title for Question 3:** Query Equalization ( Straightforward condition and Negation condition)

**Solution:**

```
SELECT FirstName, LastName
FROM Students
WHERE Age > 18;
SELECT FirstName, LastName
FROM Students
WHERE NOT Age <=18;
```

**TestCases:**

S.No	Inputs	Outputs
1		FirstName   LastName -----   ----- David   Wilson FirstName   LastName -----   ----- David   Wilson
2		
3		
4		
5		
6		

**White List:**

**Black List:**

**Question 4:**

From the following tables write a SQL query to find the salesperson and customer who reside in the same city. Return Salesman, cust\_name and city.

**Table Structure:**

- salesman\_id, name, city, commission

**Input table :**

Salesman			
salesman_id	name	city	commission
5001	James Hoog	New York	0.15
5002	Nail Knite	Paris	0.13
5003	Pit Alex	London	0.13
5004	Mc Lyon	Paris	0.14
5007	Paul Adam	Rome	0.13
5009	Leamon Han	San Jose	0.12

Customer				
customer_id	cust_name	city	grade	salesman_id
3002	Nick Rimando	New York	300	5001
3007	Brad Davis	New York	200	5001
3006	Orlham Dal	California	200	5002
3008	Julian Green	London	300	5002
3004	Fabian Johnson	Paris	300	5004
3009	Geoff Camaron	Berlin	300	5003
3003	Ray Abbar	Moscow	200	5007
3001	Brad Guzan	London		5005

## Title for Question 4: SQL Query to Retrieve Salespeople and Customers in the Same City

### Solution:

```
SELECT salesman.name AS "Salesman",
customer.cust_name, customer.city
FROM salesman, customer
WHERE salesman.city=customer.city;
```

### TestCases:

S.No	Inputs	Outputs
1		Salesman   cust_name   city -----   -----   ----- Pit Alex   Brad Guzan   London James Hoog   Nick Rimando   New York Mc Lyon   Fabian Johnson   Paris Nail Knite   Fabian Johnson   Paris James Hoog   Brad Davis   New York Pit Alex   Julian Green   London
2		
3		
4		
5		
6		

### White List:

### Black List:

### Question 5:

From the following tables write a SQL query to find those orders where the order amount exists between 500 and 2000. Return ord\_no, purch\_amt, cust\_name, city.

### Input Structure :

ord\_no INT PRIMARY KEY,

purch\_amt DECIMAL(10, 2),  
  
ord\_date DATE,  
  
customer\_id INT,  
  
salesman\_id INT

Input table :

ord_no	purch_amt	ord_date	customer_id	salesman_id
70001	156.5	2012-10-03	3001	5001
70009	270.45	2012-09-10	3001	5005
70002	45.56	2012-10-05	3002	5001
70004	16.5	2012-08-17	3009	5003
70007	948.5	2012-09-10	3005	5002
70006	2400.6	2012-07-27	3007	5001
70008	6760	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70003	2480.4	2012-10-10	3009	5003
70005	250.45	2012-06-27	3008	5002
70011	76.29	2012-08-17	3003	5007
70013	3045.4	2012-04-25	3002	5001

customer_id	cust_name	city	grade	salesman_id
3001	Nick Bonardo	New York	100	5001
3007	Brent Dean	New York	200	5001
3005	Graham Zusi	California	200	5002
3008	Julian Green	London	300	5002
3004	Fabian Johnson	Paris	300	5006
3009	Oswald Cavaretti	Berlin	100	5003
3003	Jerry Altobelli	Moscow	200	5007
3001	Brent Dean	London		5005

Title for Question 5: SQL Query to Retrieve Orders with Purchase Amount Between 500 and 2000, Including Customer Details

Solution:

```
SELECT  a.ord_no,a.purch_amt, b.cust_name,b.city  FROM orders a,customer
WHERE  a.customer_id=b.customer_id
AND a.purch_amt BETWEEN 500 AND 2000;
```

TestCases:

S.No	Inputs	Outputs
1		ord_no   purch_amt   cust_name   city -----   -----   -----   ----- 70007   948.50   Graham Zusi   California 70010   1983.43   Fabian Johnson   Paris
2		
3		
4		
5		
6		

White List:

Black List:

## Question 6:

From the following tables write a SQL query to calculate the average price of items of each company. Return average value and company name.

### Input Table :

COM_ID	COM_NAME
9	Samsung
12	iBall
13	Epson
14	Zebronics
15	Asus
16	Frontech

PRO_ID	PRO_NAME	PRO_PRICE	PRO_COMP
101	Monitor Stand	3200	15
102	Key Board	450	16
103	DP drive	250	14
104	Scanner	500	16
105	Monitor	5000	9
106	DVD drive	900	12
107	CD drive	800	12
108	Printer	2400	13
109	Refill cartridge	350	13
110	Mouse	250	12

**Title for Question 6:** SQL Query to Calculate Average Product Prices by Company Name

### Solution:

```
SELECT AVG(pro_price), company_mast.com_name
FROM item_mast INNER
JOIN company_mast
ON item_mast.pro_com= company_mast.com_id
GROUP BY company_mast.com_name;
```

### TestCases:

S.No	Inputs	Outputs
1		AVG(pro_price)   com_name -----   ----- 3200.000000   Asus 500.000000   Frontech 250.000000   Zebronics 5000.000000   Samsung 650.000000   iBall 1475.000000   Epson
2		
3		
4		
5		
6		

### White List:

### Black List:

---