

## 1. SELECT QUERY

The screenshot shows an Excel spreadsheet with a dataset of employee information. The columns are labeled A through W, and the rows are numbered 1 through 27. The data includes Education, Joining Year, City, Payment Tier, Age, Gender, Ever Bench, Experience, and Leave or Not.

Education	Joining Year	City	Payment Tier	Age	Gender	Ever Bench	Experience	Leave or Not
Bachelors	2013	Pune	1	28	Female	No	3	1
Bachelors	2017	Bangalore	1	29	Male	No	3	0
Bachelors	2014	Bangalore	1	30	Female	No	3	0
Bachelors	2014	Pune	1	22	Female	No	0	1
Bachelors	2014	Bangalore	1	27	Male	No	5	0
Bachelors	2014	Bangalore	1	23	Female	No	1	0
PHD	2015	New Delhi	1	37	Male	No	1	0
Masters	2015	Pune	1	28	Female	No	3	1
Bachelors	2014	Pune	1	26	Female	No	4	1
Bachelors	2012	Bangalore	1	28	Female	No	3	0
Masters	2013	Pune	1	24	Male	No	2	1
Masters	2017	New Delhi	1	25	Female	No	3	1
Masters	2013	Bangalore	1	25	Male	No	3	1
Bachelors	2015	Pune	1	24	Female	Yes	2	1
Bachelors	2016	Bangalore	1	28	Female	No	2	0
Bachelors	2013	New Delhi	1	27	Female	No	5	0
Bachelors	2012	Pune	1	26	Male	No	4	0
Masters	2016	New Delhi	1	27	Male	No	5	0
Bachelors	2015	Pune	1	25	Female	No	3	1
Bachelors	2017	Bangalore	1	25	Male	No	3	0
Bachelors	2016	Pune	1	26	Female	No	4	1
Masters	2017	Bangalore	1	26	Female	No	4	1
Bachelors	2017	New Delhi	1	28	Female	No	2	0
Masters	2015	New Delhi	1	27	Male	No	5	0
Bachelors	2014	Pune	1	26	Female	No	4	1
Bachelors	2015	Pune	1	26	Female	No	4	1

## 2.AVERAGE EXPERIENCE BY GROUP BY CLAUSE

The screenshot shows the MySQL Workbench interface. The SQL editor contains three queries. Query 1 is a simple SELECT statement. Query 2 is a SELECT statement with a GROUP BY clause. Query 3 is an INNER JOIN statement. The Results tab shows the output of Query 2, which is a table with two columns: Education and AvgExperience.

```
31 SELECT * FROM Employee WHERE PaymentTier = 1;
32
33 -- Query 2: Average experience grouped by education
34 SELECT Education, AVG(ExperienceInCurrentDomain) AS AvgExperience
35 FROM Employee
36 GROUP BY Education
37 ORDER BY AvgExperience DESC;
38
39 -- Query 3: INNER JOIN Employee with Department
40 SELECT E.Education, E.City, D.DepartmentName, E.Age
41 FROM Employee E
42 INNER JOIN Department D ON E.City = D.City
43 ORDER BY E.City;
```

Education	AvgExperience
Bachelors	2.9106
PHD	2.9106
Masters	2.8843

### 3.SUBQUERY

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following queries:

```
41 FROM Employee E
42 INNER JOIN Department D ON E.City = D.City
43 ORDER BY E.City;

-- Query 4: Subquery - Employees with more experience than the average
44
45 * SELECT * FROM Employee
46 WHERE ExperienceInCurrentDomain > (
47 SELECT AVG(ExperienceInCurrentDomain) FROM Employee);
48
49
50 -- Query 5: Create a View for employees who were never benched
51 CREATE VIEW AvgExpBenchTreshold AS
```

The Result Grid shows the following data:

Education	JoiningYear	City	PaymentTier	Age	Gender	EverBenchd	ExperienceInCurrentDomain	LeaveOrNot
Bachelors	2017	Pune	2	27	Female	No	5	1
Bachelors	2013	Bangalore	3	27	Male	No	5	1
Bachelors	2012	Bangalore	3	25	Male	No	3	0
Bachelors	2014	Pune	1	26	Female	No	4	1
Masters	2013	New Delhi	3	26	Male	No	4	0
Bachelors	2017	Bangalore	3	26	Female	No	4	0
Bachelors	2017	New Delhi	2	25	Male	No	3	0
Bachelors	2012	Bangalore	1	28	Female	No	3	0
PhD	2015	New Delhi	3	26	Male	No	4	0
Bachelors	2015	New Delhi	3	25	Female	No	3	0
Bachelors	2012	Bangalore	3	26	Male	No	4	1

The Action Output shows the following messages:

```
23 14:30:29 SELECT E.Education, E.City, D.DepartmentName, E.Age FROM Employee E INNER JOIN Department D ON E... 0 row(s) returned
24 14:34:14 SELECT E.Education, E.City, D.DepartmentName, E.Age FROM Employee E INNER JOIN Department D ON T... 0 row(s) returned
25 14:34:44 SELECT * FROM Employee WHERE ExperienceInCurrentDomain > ( SELECT AVG(ExperienceInCurrentDom... 1000 row(s) returned
```

### 4.INNER JOIN

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following queries:

```
35 FROM Employee
36 GROUP BY Education
37 ORDER BY AvgExperience DESC;

-- Query 3: INNER JOIN Employee with Department
38
39 * SELECT E.Education, E.City, D.DepartmentName, E.Age
40 FROM Employee E
41 INNER JOIN Department D ON E.City = D.City
42 ORDER BY E.City;

-- Query 4: Subquery - Employees with more experience than the average
```

The Result Grid shows the following data:

Education	City	DepartmentName	Age
-----------	------	----------------	-----

The Action Output shows the following messages:

```
36 14:43:14 SELECT E.Education, E.City, D.DepartmentName FROM Employee E LEFT JOIN Department D ON E.City = D... 1000 row(s) returned
37 14:43:35 SELECT City, SUM(ExperienceInCurrentDomain) AS TotalExperience FROM Employee GROUP BY City ORDER... 3 row(s) returned
38 14:53:33 SELECT E.Education, E.City, D.DepartmentName, E.Age FROM Employee E INNER JOIN Department D ON E... 0 row(s) returned
```

## 5.VIEW

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following queries:

```

47 WHERE ExperienceInCurrentDomain > (
48   SELECT AVG(ExperienceInCurrentDomain) FROM Employee);
49
50 -- Query 5: Create a View for employees who were never benched
51 * CREATE VIEW ActiveEmployees AS
52   SELECT * FROM Employee WHERE EverBenched = 'No';
53 * select * from ActiveEmployees;
54
55 -- Query 6: Create index on ExperienceInCurrentDomain for performance
56 * CREATE INDEX idx_experience ON Employee(ExperienceInCurrentDomain);

```

The Result Grid shows the data for the 'employee' table:

Education	JoiningYear	City	PaymentTier	Age	Gender	EverBenched	ExperienceInCurrentDomain	LeaveOrNot
Bachelors	2017	Bangalore	3	34	Male	No	0	0
Bachelors	2013	Pune	1	28	Female	No	3	1
Bachelors	2014	New Delhi	3	38	Female	No	2	0
Masters	2016	Bangalore	3	27	Male	No	5	1
Bachelors	2016	Bangalore	3	22	Male	No	0	0
Bachelors	2015	New Delhi	3	38	Male	No	0	0
Bachelors	2016	Bangalore	3	34	Female	No	2	1
Bachelors	2016	Pune	3	23	Male	No	1	0
Masters	2017	New Delhi	2	37	Male	No	2	0
Masters	2012	Bangalore	3	27	Male	No	5	1
Bachelors	2016	Pune	3	34	Male	No	3	0

The Action Output shows the execution of the queries:

```

26 14:35:17 CREATE VIEW ActiveEmployees AS SELECT * FROM Employee WHERE EverBenched = 'No'
0 row(s) affected
0.000 sec
27 14:36:52 select * from ActiveEmployees LIMIT 0, 1000
Error Code: 1146, Table 'employee_sql.activeemployees' doesn't exist
0.000 sec
28 14:37:14 select * from ActiveEmployees LIMIT 0, 1000
1000 row(s) returned
0.000 sec / 0.000 sec

```

## 6.INDEX

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following queries:

```

53 * select * from ActiveEmployees;
54
55 -- Query 6: Create index on ExperienceInCurrentDomain for performance
56 * CREATE INDEX idx_experience ON Employee(ExperienceInCurrentDomain);
57 * SHOW INDEX FROM EMPLOYEE;
58
59 -- Query 7: ORDER BY - List employees sorted by age descending
60 * SELECT * FROM Employee
61   ORDER BY Age DESC;
62
63 -- Query 8: TEST ORDER WITH ALL documents and their experience (if any)

```

The Result Grid shows the data for the 'employee' table:

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
employee	1	idx_experience	1	ExperienceInCurrentDomain	A	8			YES	BTREE			YES	

The Action Output shows the execution of the queries:

```

29 14:37:50 CREATE INDEX idx_experience ON Employee(ExperienceInCurrentDomain)
0 row(s) affected, Records: 0 Duplicates: 0 Warnings: 0
0.047 sec
30 14:38:31 SELECT * FROM Employee ORDER BY Age DESC LIMIT 0, 1000
Error Code: 1146, Table 'employee_sql.idx_experience' doesn't exist
0.000 sec
31 14:41:11 SHOW INDEX FROM EMPLOYEE
1 row(s) returned
0.000 sec / 0.000 sec

```

## 7.ORDERBY

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following queries:

```
53 select *from ActiveEmployees;
54
55 -- Query 6: Create index on ExperienceInCurrentDomain for performance
56 CREATE INDEX idx_experience ON Employee(ExperienceInCurrentDomain);
57 SHOW INDEX FROM EMPLOYEE;
58
59 -- Query 7: ORDER BY - List employees sorted by age descending
60 SELECT * FROM Employee
61 ORDER BY Age DESC;
62
```

The 'Result Grid' shows the output of the first query, displaying a table with columns: Table, Non\_unique, Key\_name, Seq\_in\_index, Column\_name, Collation, Cardinality, Sub\_part, Packed, Null, Index\_type, Comment, Index\_comment, Visible, and Expression. The table 'employee' has a non-unique index 'idx\_experience' on the 'ExperienceInCurrentDomain' column.

The 'Action Output' pane shows the execution results of the queries:

#	Time	Action	Message	Duration / Fetch
29	14:37:50	CREATE INDEX idx_experience ON Employee(ExperienceInCurrentDomain)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.047 sec
30	14:38:31	SELECT *FROM idx_experience LIMIT 0, 1000	Error Code: 1146. Table 'employee_idx_idx_experience' doesn't exist	0.000 sec
31	14:41:11	SHOW INDEX FROM EMPLOYEE	1 row(s) returned	0.000 sec / 0.000 sec

## 8.LEFT OUTER JOIN

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following queries:

```
62 ORDER BY Age DESC;
63
64 -- Query 8: LEFT OUTER JOIN - All dept Workbench will automatically add the LIMIT clause with the configured number of rows to SELECT queries.
65 SELECT D.DepartmentName, E.Education, E.Gender, E.City
66 FROM Department D
67 LEFT OUTER JOIN Employee E ON D.City = E.City
68 ORDER BY D.DepartmentName;
69
70 -- Query 9: RIGHT OUTER JOIN simulated
71 SELECT E.Education, E.City, D.DepartmentName
72 FROM Employee E
73 RIGHT OUTER JOIN Department D ON E.City = D.City
74 ORDER BY D.DepartmentName;
```

The 'Result Grid' shows the output of the first query, displaying a table with columns: DepartmentName, Education, Gender, and City.

The 'Action Output' pane shows the execution results of the queries:

#	Time	Action	Message	Duration / Fetch
37	14:43:35	SELECT City, SUM(ExperienceInCurrentDomain) AS TotalExperience FROM Employee GROUP BY City ORDER...	3 row(s) returned	0.000 sec / 0.000 sec
38	14:53:33	SELECT E.Education, E.City, D.DepartmentName, E.Age FROM Employee E INNER JOIN Department D ON E...	0 row(s) returned	0.000 sec / 0.000 sec
39	14:57:01	SELECT D.DepartmentName, E.Education, E.Gender, E.City FROM Department D LEFT OUTER JOIN Employee...	0 row(s) returned	0.000 sec / 0.000 sec

## 9. RIGHT OUTER JOIN

The screenshot shows the MySQL Workbench interface with a SQL query window. The query is as follows:

```
65 SELECT D.DepartmentName, E.Education, E.Gender, E.City
66 FROM
67 LEFT OUTER JOIN Employee E ON D.City = E.City
68 ORDER BY D.DepartmentName;
69
70 -- Query 9: RIGHT OUTER JOIN simulated
71 SELECT E.Education, E.City, D.DepartmentName
72 FROM Employee E
73 LEFT JOIN Department D ON E.City = D.City;
```

The Result Grid shows the output of the query:

Education	City	DepartmentName
Bachelors	Bangalore	0000
Bachelors	Pune	0000
Bachelors	New Delhi	0000
Masters	Bangalore	0000
Masters	Pune	0000
Bachelors	Bangalore	0000
Bachelors	New Delhi	0000
Bachelors	Bangalore	0000
Bachelors	Pune	0000
Masters	New Delhi	0000
Masters	Bangalore	0000

The Action Output window shows the execution details:

#	Time	Action	Message	Duration / Fetch
38	14:53:33	SELECT E.Education, E.City, D.DepartmentName, E.Age FROM Employee E INNER JOIN Department D ON E...	0 row(s) returned	0.000 sec / 0.000 sec
39	14:57:01	SELECT D.DepartmentName, E.Education, E.Gender, E.City FROM Department D LEFT OUTER JOIN Employee...	0 row(s) returned	0.000 sec / 0.000 sec
40	14:58:41	SELECT E.Education, E.City, D.DepartmentName FROM Employee E LEFT JOIN Department D ON E.City = D...	1000 row(s) returned	0.000 sec / 0.000 sec

## 10.SUM

The screenshot shows the MySQL Workbench interface with a SQL query window. The query is as follows:

```
67 LEFT OUTER JOIN Employee E ON D.City = E.City
68 ORDER BY D.DepartmentName;
69
70 -- Query 9: RIGHT OUTER JOIN simulated (SQLite workaround)
71 SELECT E.Education, E.City, D.DepartmentName
72 FROM Employee E
73 LEFT JOIN Department D ON E.City = D.City;
74
75 -- Query 10: SUM - Total experience of employees grouped by city
76 SELECT City, SUM(ExperienceInCurrentDomain) AS TotalExperience
77 FROM Employee;
```

The Result Grid shows the output of the query:

City	TotalExperience
Bangalore	6516
Pune	3666
New Delhi	3338

The Action Output window shows the execution details:

#	Time	Action	Message	Duration / Fetch
35	14:42:41	SELECT D.DepartmentName, E.Education, E.Gender, E.City FROM Department D LEFT OUTER JOIN Employee...	0 row(s) returned	0.000 sec / 0.000 sec
36	14:43:14	SELECT E.Education, E.City, D.DepartmentName FROM Employee E LEFT JOIN Department D ON E.City = D...	1000 row(s) returned	0.000 sec / 0.000 sec
37	14:43:35	SELECT City, SUM(ExperienceInCurrentDomain) AS TotalExperience FROM Employee GROUP BY City ORDER...	3 row(s) returned	0.000 sec / 0.000 sec