

Hexaware Technologies LTD  
Foundation Training Program (Java Batch 2)

Trainer: Mr. Veerababu

Trainee: Suraj Kumar

Date: 29/11/2023

## SQL Internal Assessment

1. Write a mysql statement to find the concatenated first\_name, last\_name where the age of the employee is greater than 30.

Suppose the employee table is -

first_name	last_name	age	dept
Mesa	Loop	30	Acct
Smith	Oak	27	Dev1
John	Jorz	37	QA
Hary	Gaga	32	QA

Solution:

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

```
1 • create database hexa_assignment;
2 • use hexa_assignment;
3 -- Given Table
4 • create table tab1(first_name varchar(10),last_name varchar(10),age int,dept varchar(10));
5 • insert into tab1 values ('Mesa','Loop',30,'Acct'),('Smith','Oak',27,'Dev1'),('John','Jorz',37,'QA'),('Hary','Gaga',32,'QA');
6 # 1. Write a mysql statement to find the concatenated first_name, last_name where the age of the employee is greater than 30.
7 • select concat(first_name,' ',last_name) as Full_name, age from tab1 where age>30;
8
9
10 # 2. Write a mysql statement to get user, current date and mysql version.
11 • select user(),current_date(),version();
```

The Results Grid shows the output of the first query:

Full_name	age
Ratnesh Joez	37
Shivam Bak	37
Ratnesh Joez	37
Ritik Joez	37

The Output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
100	00:33:30	select distinct salary from emp order by salary limit 3, 1	1 row(s) returned	0.000 sec / 0.000 sec
101	00:35:01	select concat(first_name,' ',last_name) as Full_name, age from tab1 where age>30 LIMIT 0, 1000	4 row(s) returned	0.015 sec / 0.000 sec

## 2. Write a mysql statement to get user, current date and mysql version.

### Solution:

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

```
6 # 1. Write a mysql statement to find the concatenated first_name, last_name where the age of the employee is greater than 30.
7 select concat(first_name, ' ', last_name) as Full_name, age from tab1 where age > 30;
8
9
10 # 2. Write a mysql statement to get user, current date and mysql version.
11 select user(), current_date(), version();
12
13
14 -- Given Table
15 create table item (item_id int, item_price int, item_name varchar(50));
16 insert into item values (1, 19, 'Laptop'), (2, 9, 'Mouse'), (3, 49, 'Printer'), (4, 29, 'Headphones'), (5, 5, 'USB Drive'), (6, 79, 'External Hard Drive');
17 select * from item;
```

The Result Grid shows the output of the second query:

user()	current_date()	version()
root@localhost	2023-11-30	8.0.26

The Output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
101	00:35:01	select concat(first_name, ' ', last_name) as Full_name, age from tab1 where age > 30 LIMIT 0, 1000	4 row(s) returned	0.015 sec / 0.000 sec
102	00:35:58	select user(), current_date(), version() LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

## 3. Write a mysql statement to get item id, item, price of the most expensive item.

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

```
13
14 -- Given Table
15 create table item (item_id int, item_price int, item_name varchar(50));
16 insert into item values (1, 19, 'Laptop'), (2, 9, 'Mouse'), (3, 49, 'Printer'), (4, 29, 'Headphones'), (5, 5, 'USB Drive'), (6, 79, 'External Hard Drive');
17 select * from item;
18 # 3. Write a mysql statement to get item id, item, price of the most expensive item.
19 select item_id, item_name as item, item_price as price from item order by item_price desc limit 1;
20
21
22 # 4. Write a mysql statement to create a new user and set a password and privileges for an existing database.
23 -- Existing Database - hexa_assignment
24 create database hexa_assignment;
25 create user 'hexa'@'localhost' identified by 'hexa123456789';
26 grant all privileges on hexa_assignment.* to 'hexa'@'localhost';
27 flush privileges;
```

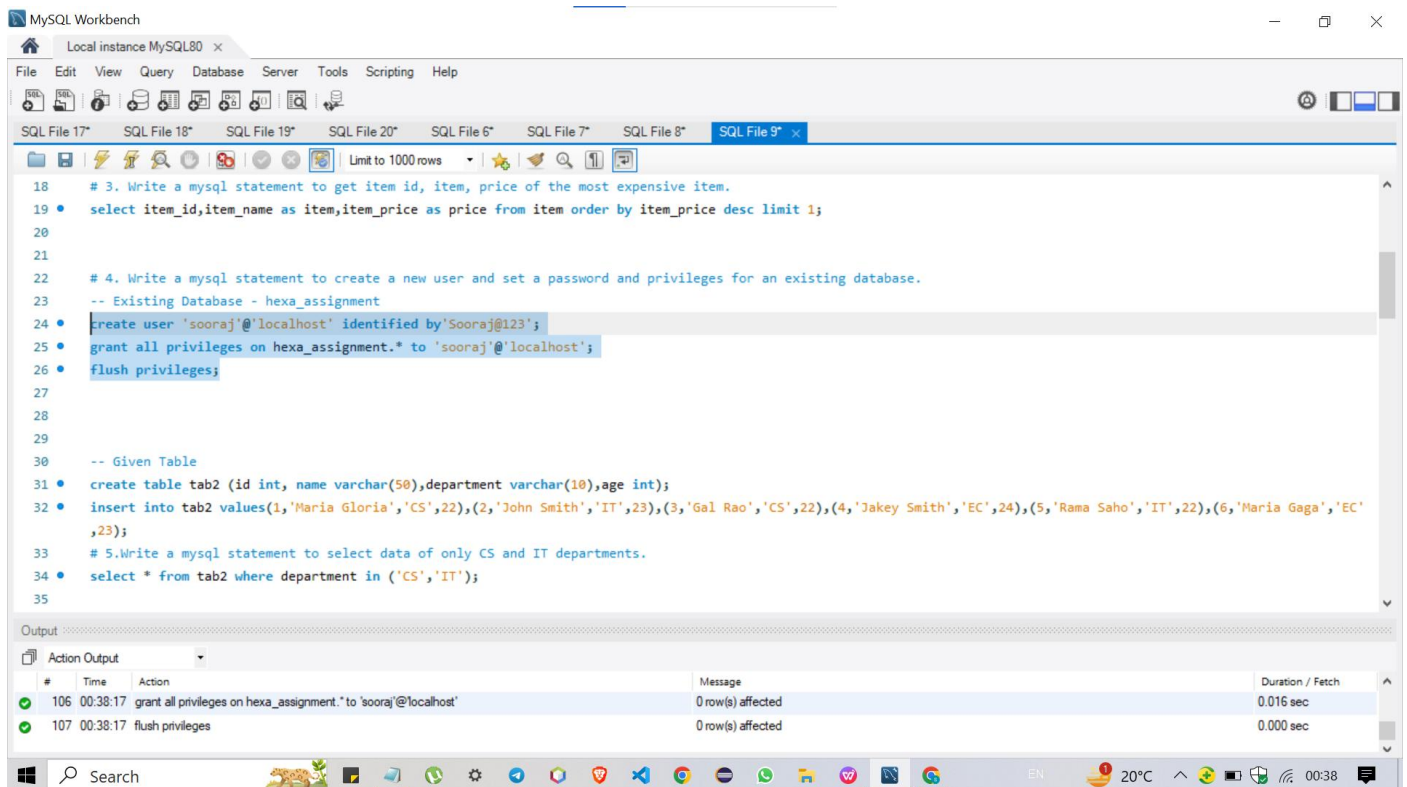
The Result Grid shows the output of the third query:

item_id	item	price
6	External Hard Drive	79

The Output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
102	00:35:58	select user(), current_date(), version() LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
103	00:37:02	select item_id, item_name as item, item_price as price from item order by item_price desc limit 1	1 row(s) returned	0.016 sec / 0.000 sec

4. Write a mysql statement to create a new user and set a password and privileges for an existing database.



The screenshot shows the MySQL Workbench interface with a local instance of MySQL 8.0. The SQL Editor contains several queries. Query 4, which is highlighted, is for creating a new user and setting privileges. The Output window at the bottom shows the execution results for two queries: granting privileges and flushing them.

```
18 # 3. Write a mysql statement to get item id, item, price of the most expensive item.
19 • select item_id,item_name as item,item_price as price from item order by item_price desc limit 1;
20
21
22 # 4. Write a mysql statement to create a new user and set a password and privileges for an existing database.
23 -- Existing Database - hexa_assignment
24 • create user 'sooraj'@'localhost' identified by 'Sooraj@123';
25 • grant all privileges on hexa_assignment.* to 'sooraj'@'localhost';
26 • flush privileges;
27
28
29
30 -- Given Table
31 • create table tab2 (id int, name varchar(50),department varchar(10),age int);
32 • insert into tab2 values(1,'Maria Gloria','CS',22),(2,'John Smith','IT',23),(3,'Gal Rao','CS',22),(4,'Jakey Smith','EC',24),(5,'Rama Saho','IT',22),(6,'Maria Gaga','EC',23);
33 # 5. Write a mysql statement to select data of only CS and IT departments.
34 • select * from tab2 where department in ('CS','IT');
35
```

Output

#	Time	Action	Message	Duration / Fetch
✓ 106	00:38:17	grant all privileges on hexa_assignment.* to 'sooraj'@'localhost'	0 row(s) affected	0.016 sec
✓ 107	00:38:17	flush privileges	0 row(s) affected	0.000 sec

5. Write a mysql statement to select data of only CS and IT departments.

Suppose the table is -

id	name	department	age
1	Maria Gloria	CS	22
2	John Smith	IT	23
3	Gal Rao	CS	22
4	Jakey Smith	EC	24
5	Rama Saho	IT	22
6	Maria Gaga	EC	23

## Solution:

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

```
-- Given Table
31 • create table tab2 (id int, name varchar(50), department varchar(10), age int);
32 • insert into tab2 values(1,'Maria Gloria','CS',22),(2,'John Smith','IT',23),(3,'Gal Rao','CS',22),(4,'Jakey Smith','EC',24),(5,'Rama Saho','IT',22),(6,'Maria Gaga','EC',23);
33 # 5. Write a mysql statement to select data of only CS and IT departments.
34 • select * from tab2 where department in ('CS','IT');
35
36
37 # 6. Write a MySQL statement to select data of all departments in descending order by age. alter
38 • select * from tab2 order by age desc;
```

The Result Grid shows the output of the query:

id	name	department	age
1	Maria Gloria	CS	22
2	John Smith	IT	23
3	Gal Rao	CS	22
5	Rama Saho	IT	22

The Action Output pane shows the following messages:

#	Time	Action	Message	Duration / Fetch
107	00:38:17	flush privileges	0 row(s) affected	0.000 sec
108	00:38:44	select * from tab2 where department in ('CS','IT') LIMIT 0, 1000	4 row(s) returned	0.015 sec / 0.000 sec

6. Write a mysql statement to select data of all departments in descending order by age.

Suppose the table is -

id	name	department	age
1	Maria Gloria	CS	22
2	John Smith	IT	23
3	Gal Rao	CS	22
4	Jakey Smith	EC	24
5	Rama Saho	IT	22
6	Maria Gaga	EC	23

## Solution:

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

```
33 # 5. Write a mysql statement to select data of only CS and IT departments.
34 • select * from tab2 where department in ('CS','IT');
35
36
37 # 6. Write a MySQL statement to select data of all departments in descending order by age. alter
38 • select * from tab2 order by age desc;
39
40
41
42 -- Given table
43 create table tab2 (id int primary key(10), department varchar(10), birth date);
```

The Result Grid shows the output of the query `select * from tab2 order by age desc`:

id	name	department	age
4	Jakey Smith	EC	24
2	John Smith	IT	23
6	Maria Gaga	EC	23
1	Maria Gloria	CS	22
3	Gal Rao	CS	22

The Action Output pane shows the execution details:

#	Time	Action	Message	Duration / Fetch
110	00:39:04	select * from tab2 order by age desc LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
111	00:39:09	select * from tab2 order by age desc LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec

7. Write a mysql statement to determine the age of each of the students.

Suppose the table is -

id	name	department	birth
1	Maria Gloria	CS	1994-03-12
2	John Smith	IT	1993-02-07
3	Gal Rao	CS	1992-09-11
4	Jakey Smith	EC	1990-08-31
5	Rama Saho	IT	1994-12-09
6	Maria Gaga	EC	1993-10-09

Solution:

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

```
-- Given table
create table tab3 (id int, name varchar(50), department varchar(10), birth date);
insert into tab3 values(1,'Maria Gloria','CS','1994-03-12'),(2,'John Smith','IT','1993-02-07'),(3,'Gal Rao','CS','1992-09-11'),(4,'Jakey Smith','EC','1990-08-31'),(5,'Rama Saho','IT','1994-12-09'),(6,'Maria Gaga','EC','1993-10-09');

# 7. Write a mysql statement to determine the age of each of the students.
select id,name,timediff(year, birth, curdate()) as age from tab3;

# 8. Write a mysql statement to retrieve name beginning with 'm'.
select name from tab3 where name like 'm%';
```

The Results window shows the output of the first query (Result 49):

id	name	age
1	Maria Gloria	29
2	John Smith	30
3	Gal Rao	31
4	Jakey Smith	33
5	Rama Saho	28
6	Maria Gaga	31

The Output window shows the execution of the queries:

#	Time	Action	Message	Duration / Fetch
111	00:39:09	select * from tab2 order by age desc LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
112	00:39:39	select id,name,timediff(year, birth, curdate()) as age from tab3 LIMIT 0, 1000	12 row(s) returned	0.000 sec / 0.000 sec



8. Write a mysql statement to retrieve name beginning with 'm'.

Suppose the table is -

id	name	department	birth
1	Maria Gloria	CS	1994-03-12
2	John Smith	IT	1993-02-07
3	Gal Rao	CS	1992-09-11
4	Jakey Smith	EC	1990-08-31
5	Rama Saho	IT	1994-12-09
6	Maria Gaga	EC	1993-10-09

+-----+-----+-----+-----+

## Solution:

The screenshot shows the MySQL Workbench interface. The SQL Editor contains the following code:

```
46 • select id,name,timestampdiff(year, birth, curdate()) as age from tab3;
47
48 # 8. Write a mysql statement to retrieve name beginning with 'm'.
49 • select name from tab3 where name like 'm%';
50
51
52 -- Given Table
53 # Employee Table
54 • create table tab4 (id int, name varchar(50),dept_id int,birth date);
55 • insert into tab4 values(1,'Maria Gloria',2,'1994-03-12'),(2,'John Smith',1,'1993-02-07'),(3,'Gal Rao',4,'1992-09-11'),(4,'Jakey Smith',2,'1990-08-31'),(5,'Rama Saho',1,'1994-12-09'),(6,'Maria Gaga',4,'1993-10-09');
```

The Results window shows the output of the query `select name from tab3 where name like 'm%';`. The results are:

name
Maria Gloria
Maria Gaga
Maria Gloria
Maria Gaga

The Output window shows the execution of the queries:

#	Time	Action	Message	Duration / Fetch
112	00:39:39	select id,name,timestampdiff(year, birth, curdate()) as age from tab3 LIMIT 0, 1000	12 row(s) returned	0.000 sec / 0.000 sec
113	00:40:01	select name from tab3 where name like 'm%' LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec

9. Write a mysql statement to find the name, birth, department name, department block from the given tables.

id	name	dept_id	birth
1	Maria Gloria	2	1994-03-12
2	John Smith	1	1993-02-07
3	Gal Rao	4	1992-09-11
4	Jakey Smith	2	1990-08-31
5	Rama Saho	1	1994-12-09
6	Maria Gaga	4	1993-10-09

dept_id	dept_name	dept_block
1	Computer Science	B-Block
2	Information Technology	C-Block
3	Mechanical	A-Block
4	Electronic Communication	D-Block

## Solution:

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

SQL File 17\* SQL File 18\* SQL File 19\* SQL File 20\* SQL File 6\* SQL File 7\* SQL File 8\* SQL File 9\* x

Limit to 1000 rows

```
-- Given Table
# Employee Table
create table tab4 (id int, name varchar(50),dept_id int,birth date);
insert into tab4 values(1,'Maria Gloria',2,'1994-03-12'),(2,'John Smith',1,'1993-02-07'),(3,'Gal Rao',4,'1992-09-11'),(4,'Jakey Smith',2,'1990-08-31'),(5,'Rama Saho',1,'1994-12-09'),(6,'Maria Gaga',4,'1993-10-09');
# Department Table
create table dept(dept_id int,dept_name varchar(50),dept_block varchar(10));
insert into dept values(1,'Computer Science','B-Block'),(2,'Information Technology','C-Block'),(3,'Mechanical','A-Block'),(4,'Electronic Communication','D-Block');
# 9. Write a mysql statement to find the name, birth, department name, department block from the given tables.
select e.name,e.birth,d.dept_name,d.dept_block from tab4 e left join dept d on e.dept_id = d.dept_id;
-- select name,birth,dept_name,dept_block from tab4 left join dept on tab4.dept_id = dept.dept_id;
```

Result Grid

	name	birth	dept_name	dept_block
▶	Maria Gloria	1994-03-12	Information Technology	C-Block
	John Smith	1993-02-07	Computer Science	B-Block
	Gal Rao	1992-09-11	Electronic Communication	D-Block
	Jakey Smith	1990-08-31	Information Technology	C-Block
	Rama Saho	1994-12-09	Computer Science	B-Block

Result 51 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
113	00:40:01	select name from tab3 where name like 'm%' LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
114	00:40:56	select e.name,e.birth,d.dept_name,d.dept_block from tab4 e left join dept d on e.dept_id = d.dept_id LIMIT 0, 1000	6 row(s) returned	0.109 sec / 0.000 sec

Search

20°C

00:41



10. Write a mysql statement to get name of students containing exactly four characters.

Suppose the student table is -

id	name	dept_id	birth
1	Maria	2	1994-03-12
2	John	1	1993-02-07
3	Gal	4	1992-09-11
4	Jakey	2	1990-08-31
5	Rama	1	1994-12-09
6	Maria	4	1993-10-09

Solution:

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

```
61 -- select name,birth,dept_name,dept_block from tab4 left join dept on tab4.dept_id = dept.dept_id;
62
63
64 -- Given Table
65 • create table tab5 (id int, name varchar(50),dept_id int,birth date);
66 • insert into tab5 values(1,'Maria',2,'1994-03-12'),(2,'John',1,'1993-02-07'),(3,'Gal',4,'1992-09-11'),(4,'Jakey',2,'1990-08-31'),(5,'Rama',1,'1994-12-09'),(6,'Maria',4,'1993-10-09');
67
68 # 10. Write a mysql statement to get name of students containing exactly four characters.
69 • select name from tab5 where length(name) = 4;
70
71
```

The Results panel shows the output of the query:

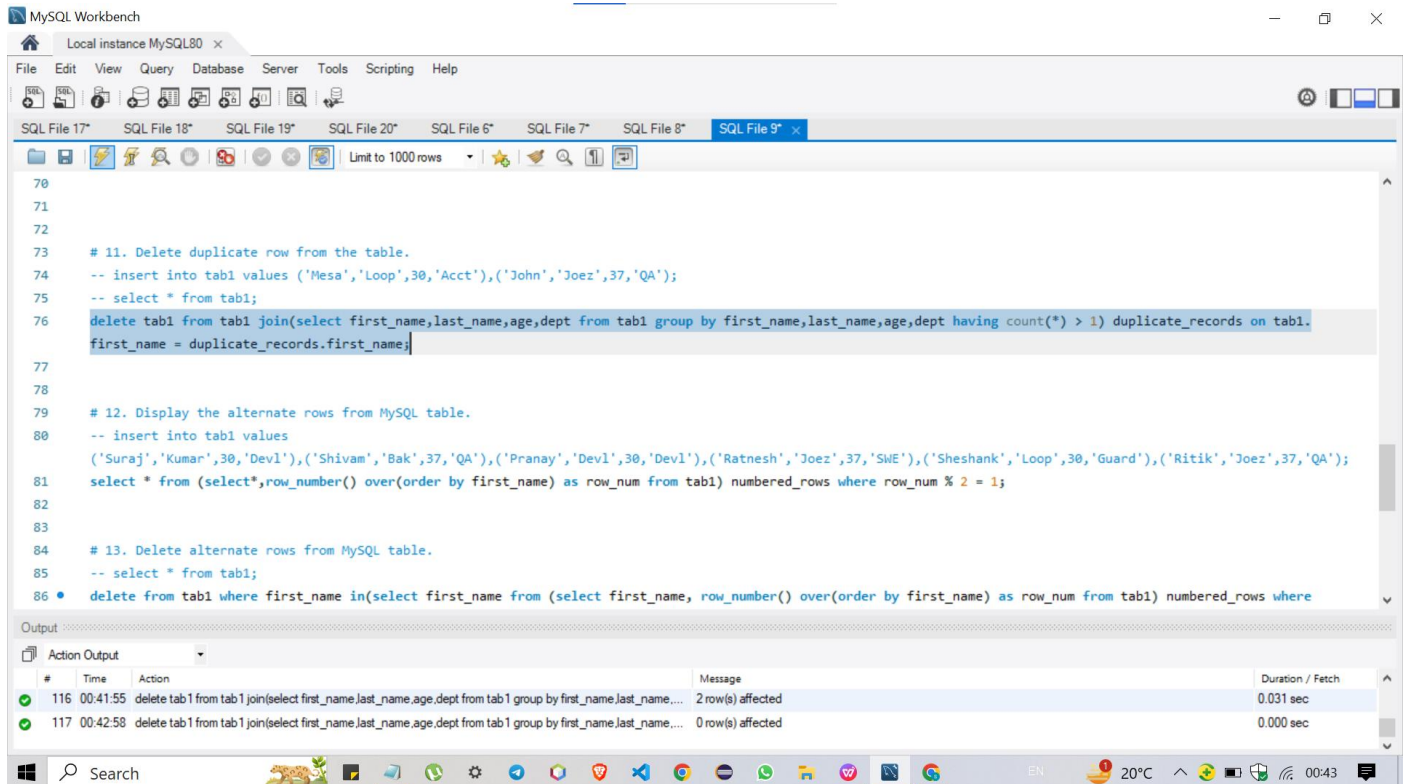
name
John
Rama

The Action Output panel shows the execution details:

#	Time	Action	Message	Duration / Fetch
114	00:40:56	select e.name,e.birth,d.dept_name,d.dept_block from tab4 e left join dept d on e.dept_id = d.dept_id LIMIT 0, 1000	6 row(s) returned	0.109 sec / 0.000 sec
115	00:41:19	select name from tab5 where length(name) = 4 LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

## 11. Delete duplicate row from the table.

### Solution:



The screenshot shows the MySQL Workbench interface with a SQL script in the editor. The script includes comments for steps 11, 12, and 13. Step 11 is the main query to delete duplicate rows. The Output pane shows the execution results for steps 116 and 117.

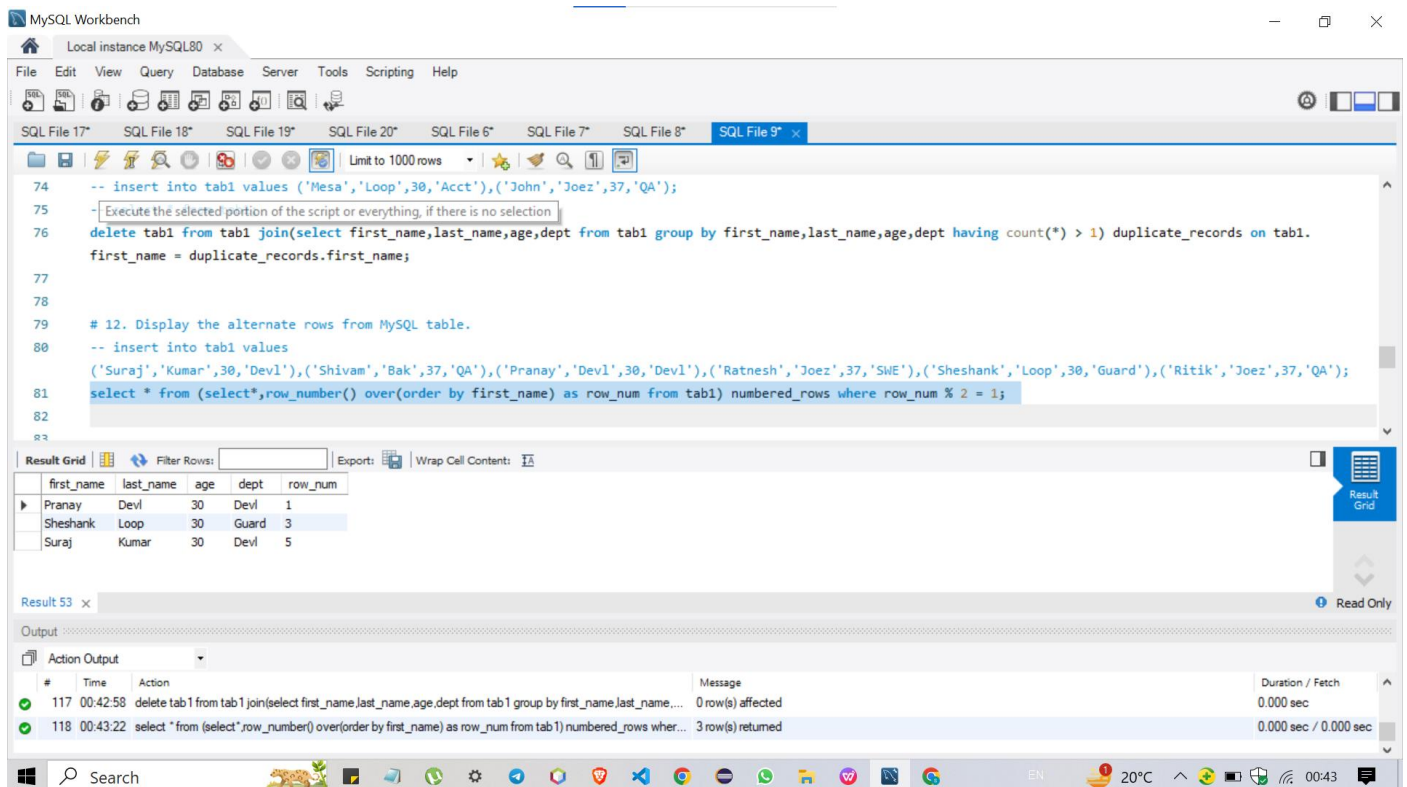
```
70
71
72
73 # 11. Delete duplicate row from the table.
74 -- insert into tab1 values ('Mesa','Loop',30,'Acct'),('John','Jozee',37,'QA');
75 -- select * from tab1;
76 delete tab1 from tab1 join(select first_name,last_name,age,dept from tab1 group by first_name,last_name,age,dept having count(*) > 1) duplicate_records on tab1.
   first_name = duplicate_records.first_name;
77
78
79 # 12. Display the alternate rows from MySQL table.
80 -- insert into tab1 values
   ('Suraj','Kumar',30,'Dev1'),('Shivam','Bak',37,'QA'),('Pranay','Dev1',30,'Dev1'),('Ratnesh','Jozee',37,'SWE'),('Sheshank','Loop',30,'Guard'),('Ritik','Jozee',37,'QA');
81 select * from (select*,row_number() over(order by first_name) as row_num from tab1) numbered_rows where row_num % 2 = 1;
82
83
84 # 13. Delete alternate rows from MySQL table.
85 -- select * from tab1;
86 • delete from tab1 where first_name in(select first_name from (select first_name, row_number() over(order by first_name) as row_num from tab1) numbered_rows where
```

Output:

#	Time	Action	Message	Duration / Fetch
116	00:41:55	delete tab1 from tab1 join(select first_name,last_name,age,dept from tab1 group by first_name,last_name,age,dept having count(*) > 1) duplicate_records on tab1.	2 row(s) affected	0.031 sec
117	00:42:58	delete tab1 from tab1 join(select first_name,last_name,age,dept from tab1 group by first_name,last_name,age,dept having count(*) > 1) duplicate_records on tab1.	0 row(s) affected	0.000 sec

## 12. Display the alternate rows from MySQL table.

### Solution:



The screenshot shows the MySQL Workbench interface with a SQL script in the editor. The script includes comments for steps 12 and 13. Step 12 is the main query to display alternate rows. The Result Grid shows the output of the query, and the Output pane shows the execution results for steps 117 and 118.

```
74 -- insert into tab1 values ('Mesa','Loop',30,'Acct'),('John','Jozee',37,'QA');
75 -- Execute the selected portion of the script or everything, if there is no selection
76 delete tab1 from tab1 join(select first_name,last_name,age,dept from tab1 group by first_name,last_name,age,dept having count(*) > 1) duplicate_records on tab1.
   first_name = duplicate_records.first_name;
77
78
79 # 12. Display the alternate rows from MySQL table.
80 -- insert into tab1 values
   ('Suraj','Kumar',30,'Dev1'),('Shivam','Bak',37,'QA'),('Pranay','Dev1',30,'Dev1'),('Ratnesh','Jozee',37,'SWE'),('Sheshank','Loop',30,'Guard'),('Ritik','Jozee',37,'QA');
81 select * from (select*,row_number() over(order by first_name) as row_num from tab1) numbered_rows where row_num % 2 = 1;
82
```

Result Grid:

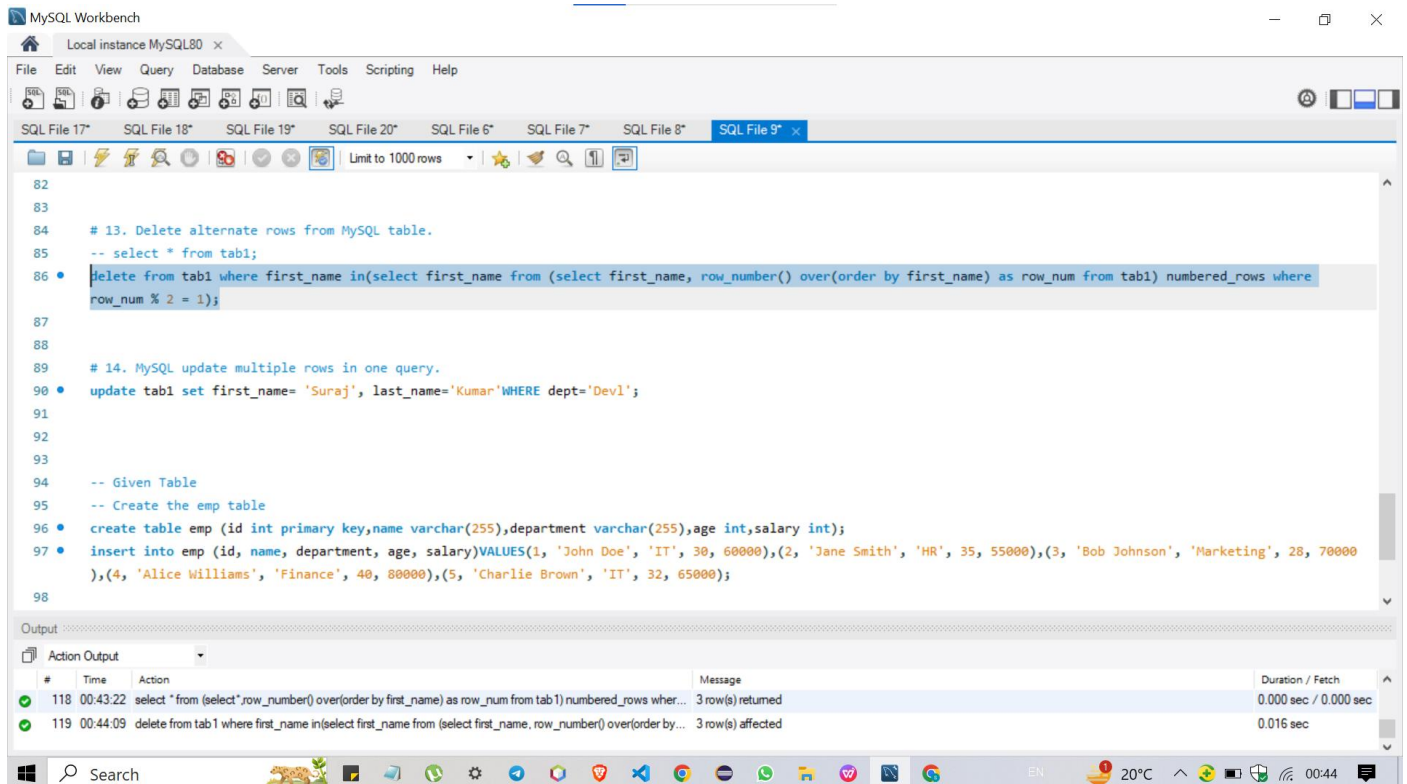
first_name	last_name	age	dept	row_num
Pranay	Dev1	30	Dev1	1
Sheshank	Loop	30	Guard	3
Suraj	Kumar	30	Dev1	5

Output:

#	Time	Action	Message	Duration / Fetch
117	00:42:58	delete tab1 from tab1 join(select first_name,last_name,age,dept from tab1 group by first_name,last_name,age,dept having count(*) > 1) duplicate_records on tab1.	0 row(s) affected	0.000 sec
118	00:43:22	select * from (select*,row_number() over(order by first_name) as row_num from tab1) numbered_rows where row_num % 2 = 1;	3 row(s) returned	0.000 sec / 0.000 sec

### 13. Delete alternate rows from MySQL table.

#### Solution:



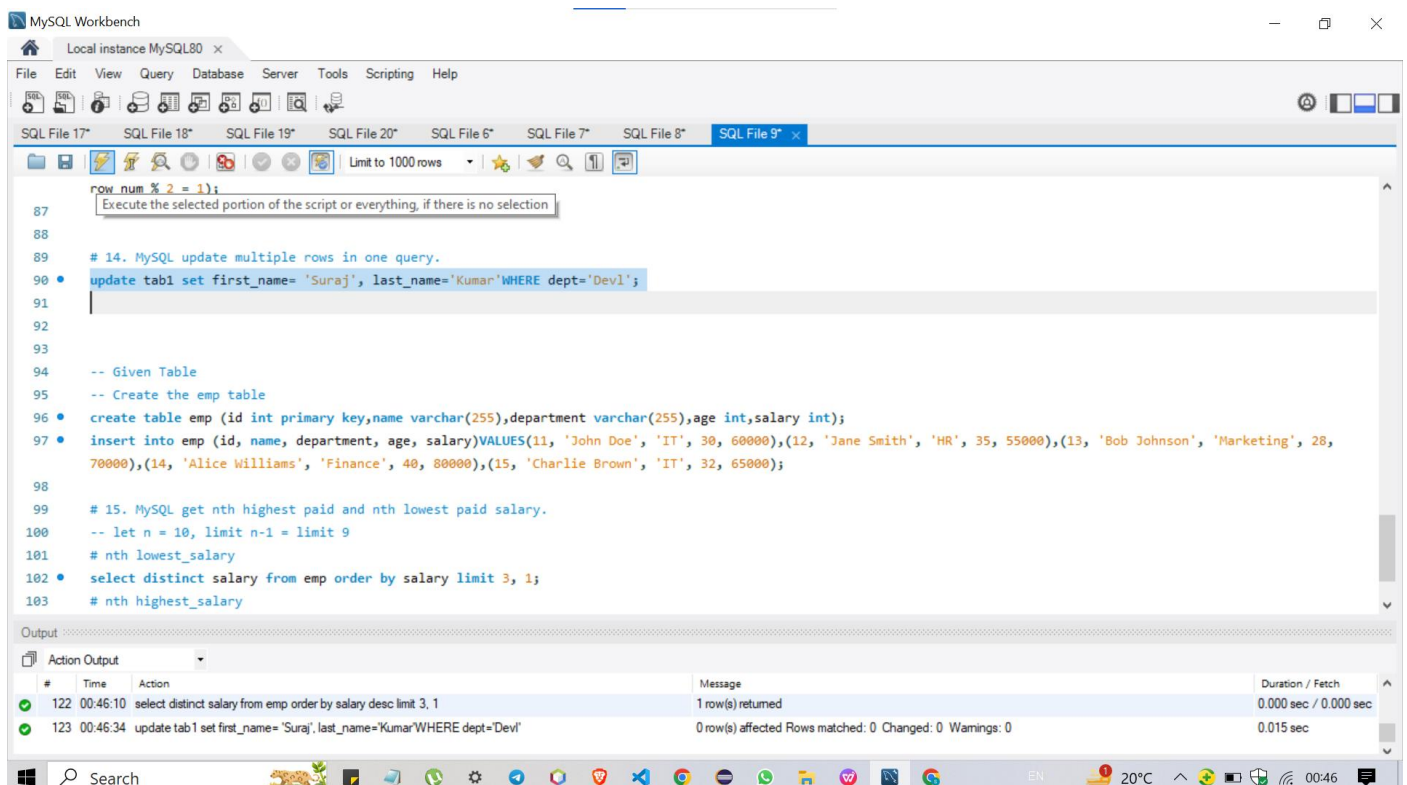
The screenshot shows the MySQL Workbench interface with a SQL script editor. The script contains two queries: Query 13, which deletes alternate rows from a table named 'tab1' using a subquery with row numbering, and Query 14, which updates multiple rows in a table named 'emp' by setting the first\_name and last\_name for specific employees. The output window shows the execution results for both queries.

```
82
83
84 # 13. Delete alternate rows from MySQL table.
85 -- select * from tab1;
86 • delete from tab1 where first_name in(select first_name from (select first_name, row_number() over(order by first_name) as row_num from tab1) numbered_rows where
row_num % 2 = 1);
87
88
89 # 14. MySQL update multiple rows in one query.
90 • update tab1 set first_name= 'Suraj', last_name='Kumar'WHERE dept='Dev1';
91
92
93
94 -- Given Table
95 -- Create the emp table
96 • create table emp (id int primary key,name varchar(255),department varchar(255),age int,salary int);
97 • insert into emp (id, name, department, age, salary)VALUES(1, 'John Doe', 'IT', 30, 60000),(2, 'Jane Smith', 'HR', 35, 55000),(3, 'Bob Johnson', 'Marketing', 28, 70000
), (4, 'Alice Williams', 'Finance', 40, 80000),(5, 'Charlie Brown', 'IT', 32, 65000);
98
```

#	Time	Action	Message	Duration / Fetch
118	00:43:22	select * from (select row_number() over(order by first_name) as row_num from tab1) numbered_rows wher...	3 row(s) returned	0.000 sec / 0.000 sec
119	00:44:09	delete from tab1 where first_name in(select first_name from (select first_name, row_number() over(order by...	3 row(s) affected	0.016 sec

### 14. MySQL update multiple rows in one query.

#### Solution:



The screenshot shows the MySQL Workbench interface with a SQL script editor. The script contains two queries: Query 14, which updates multiple rows in a table named 'emp' by setting the first\_name and last\_name for specific employees, and Query 15, which selects distinct salaries from the 'emp' table ordered by salary in descending order. The output window shows the execution results for both queries.

```
87
88
89 # 14. MySQL update multiple rows in one query.
90 • update tab1 set first_name= 'Suraj', last_name='Kumar'WHERE dept='Dev1';
91
92
93
94 -- Given Table
95 -- Create the emp table
96 • create table emp (id int primary key,name varchar(255),department varchar(255),age int,salary int);
97 • insert into emp (id, name, department, age, salary)VALUES(11, 'John Doe', 'IT', 30, 60000),(12, 'Jane Smith', 'HR', 35, 55000),(13, 'Bob Johnson', 'Marketing', 28,
70000),(14, 'Alice Williams', 'Finance', 40, 80000),(15, 'Charlie Brown', 'IT', 32, 65000);
98
99 # 15. MySQL get nth highest paid and nth lowest paid salary.
100 -- let n = 10, limit n-1 = limit 9
101 # nth lowest_salary
102 • select distinct salary from emp order by salary limit 3, 1;
103 # nth highest_salary
```

#	Time	Action	Message	Duration / Fetch
122	00:46:10	select distinct salary from emp order by salary desc limit 3, 1	1 row(s) returned	0.000 sec / 0.000 sec
123	00:46:34	update tab1 set first_name= 'Suraj', last_name='Kumar'WHERE dept='Dev1'	0 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0	0.015 sec

## 15. MySQL get nth highest paid and nth lowest paid salary.

I have created the following table according the required query:

```
create table emp (id int primary key,name varchar(255),department varchar(255),age int,salary int);
```

```
insert into emp (id, name, department, age, salary)VALUES(11, 'John Doe', 'IT', 30, 60000),(12, 'Jane Smith', 'HR', 35, 55000),(13, 'Bob Johnson', 'Marketing', 28, 70000),(14, 'Alice Williams', 'Finance', 40, 80000),(15, 'Charlie Brown', 'IT', 32, 65000);
```

Solution:

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

```
96 • create table emp (id int primary key,name varchar(255),department varchar(255),age int,salary int);
97 • insert into emp (id, name, department, age, salary)VALUES(11, 'John Doe', 'IT', 30, 60000),(12, 'Jane Smith', 'HR', 35, 55000),(13, 'Bob Johnson', 'Marketing', 28, 70000),(14, 'Alice Williams', 'Finance', 40, 80000),(15, 'Charlie Brown', 'IT', 32, 65000);
98
99 # 15. MySQL get nth highest paid and nth lowest paid salary.
100 -- let n = 10, limit n-1 = limit 9
101 # nth lowest_salary
102 • select distinct salary from emp order by salary limit 3, 1;
103 # nth highest_salary
104 • select distinct salary from emp order by salary desc limit 3, 1;
105
```

The Results tab shows the output of the first query, displaying a single row with the salary 60000.

The Output tab shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
121	00:45:16	insert into emp (id, name, department, age, salary)VALUES(11, 'John Doe', 'IT', 30, 60000),(12, 'Jane Smit...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.031 sec
122	00:46:10	select distinct salary from emp order by salary desc limit 3, 1	1 row(s) returned	0.000 sec / 0.000 sec

Thank you