

DBMS Assignment-5

By Group 3 -

18BCS089,18BCS092,18BCS095,18BCS117,18BCS041,18BCS098

1. INNER JOIN - 3 queries

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

```
1. show databases;
2. use mydb;
3. select * from tickets inner join
4. transactions on
5. tickets.transactions= transactions.transactions_id and
6. transactions.transactions_id=i;
```

The query results are displayed in a table with the following columns: Ticket_id, BookingTime, DepartureTime, Agency, Transactions, Connections, BookingClass, DepartureClass, transactions_id, Transaction_date, amount, Date. The results show three rows of data.

Ticket_id	BookingTime	DepartureTime	Agency	Transactions	Connections	BookingClass	DepartureClass	transactions_id	Transaction_date	amount	Date
1000	2020-09-08 18:30:00	2020-09-12 18:30:00	12024	1	200	4	2	1	2020-09-12 18:30:00	200	2020-09-12
10001	2020-09-08 18:30:00	2020-09-12 18:30:00	12024	1	200	4	2	1	2020-09-12 18:30:00	200	2020-09-12
10002	2020-09-08 18:30:00	2020-09-12 18:30:00	12024	1	200	4	2	1	2020-09-12 18:30:00	200	2020-09-12

The bottom panel shows the execution log with the following entries:

Time	Action	Status	Message	Duration
10:00:00	select * from tickets inner join transactions on tickets.transactions=transactions_id and transactions.transactions_id=i; LIMIT 0, 1000	Succeeded	0 rows returned	0.000 sec / 0.000 sec
10:00:01	select * from tickets inner join transactions on tickets.transactions=transactions_id and transactions.transactions_id=i; LIMIT 0, 1000	Succeeded	0 rows returned	0.000 sec / 0.000 sec
10:00:02	select * from tickets inner join transactions on tickets.transactions=transactions_id and transactions.transactions_id=i; LIMIT 0, 1000	Succeeded	0 rows returned	0.000 sec / 0.000 sec
10:00:03	select * from tickets inner join transactions on tickets.transactions=transactions_id and transactions.transactions_id=i; LIMIT 0, 1000	Succeeded	0 rows returned	0.000 sec / 0.000 sec
10:00:04	select * from tickets inner join transactions on tickets.transactions=transactions_id and transactions.transactions_id=i; LIMIT 0, 1000	Succeeded	0 rows returned	0.000 sec / 0.000 sec
10:00:05	select * from tickets inner join transactions on tickets.transactions=transactions_id and transactions.transactions_id=i; LIMIT 0, 1000	Succeeded	0 rows returned	0.000 sec / 0.000 sec

a) Inner join tickets and transactions based on transition id

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

```

1. show databases;
2. use mysql;
3. select * from tickets inner join
4. transactions on
5. tickets.Transaction= transactions.transactions_id and
6. transactions.Transaction_mode like "000";

```

The Results panel displays the following data:

Ticket_id	BoardingTime	DepartureTime	Agency	Transactions	Convenience_mode	BoardingPlace	DeparturePlace	transactions_id	Transaction_mode	amount	Date
12000	2020-09-08 12:00:00	2020-09-12 07:00:00	12004	1	200	4	2	1	0000	300	2020-09-08
12001	2020-09-08 17:00:00	2020-09-12 09:00:00	12004	1	200	4	1	1	0000	300	2020-09-08
12002	2020-09-09 12:00:00	2020-09-14 18:00:00	12004	3	200	4	3	2	0000	300	2020-09-09
12004	2020-09-11 17:00:00	2020-09-16 07:00:00	12004	3	200	2	3	2	0000	300	2020-09-11

The Event Log shows the execution of the query, including the inner join operation.

b) Inner join tickets and transactions based on mode of transition

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

```

1. show databases;
2. use mysql;
3. select * from customers_has_tickets inner join
4. transactions on
5. transactions.transactions_id = customers_has_tickets.customer_id and
6. transactions.transactions_id = customers_has_tickets.ticket_id and
7. transactions.transactions_id = customers_has_tickets.transaction_id and
8. transactions.transactions_id = customers_has_tickets.transaction_id;

```

The Results panel displays the following data:

Fullname	Phone	Amount
John Doe	123456789	300
John Doe	24681012	300

The Event Log shows the execution of the query, including the inner join operation.

c) Inner join customers and transactions based on amount transitioned by joining 2 tables in between (customers_has_tickets and tickets)

2. LEFT OUTER JOIN – 3 Queries

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

```
1 • show databases;
2 • use mydb;
3 • select * from tickets left join
4   transactions on
5   tickets.transactions= transactions.transactions_id and
6   transactions.transactions_id=;
```

The Results window displays the output of the query, showing a table with columns: Ticket_id, BookingTime, DepartureTime, Agency, Transactions, Caravanous_moss, BookingTime, DepartureTime, transactions_id, Transaction_moss, Amount, and Date. The table contains 10 rows of data.

The bottom panel shows the Query History, listing the executed queries and their execution times.

a) Left outer join tickets and transactions based on transition id

MySQL Workbench

Query 1

```

1 * show databases;
2 * use mydb;
3 * select * from tickets left join
4 transactions on
5 tickets.transaction_id and
6 transactions.transaction_mode like "paid";

```

ticket_id	transaction_id	transaction_mode	transaction_amount	transaction_date	transaction_time	transaction_status	transaction_type	transaction_note	transaction_ref
1000	2020-04-09 12:30:00	2020-04-09 12:30:00	1000	1	1000	1	1	1000	1000
1001	2020-04-09 12:30:00	2020-04-09 12:30:00	1001	1	1001	1	1	1001	1001
1002	2020-04-09 12:30:00	2020-04-09 12:30:00	1002	1	1002	1	1	1002	1002
1003	2020-04-09 12:30:00	2020-04-09 12:30:00	1003	1	1003	1	1	1003	1003
1004	2020-04-09 12:30:00	2020-04-09 12:30:00	1004	1	1004	1	1	1004	1004
1005	2020-04-09 12:30:00	2020-04-09 12:30:00	1005	1	1005	1	1	1005	1005
1006	2020-04-09 12:30:00	2020-04-09 12:30:00	1006	1	1006	1	1	1006	1006
1007	2020-04-09 12:30:00	2020-04-09 12:30:00	1007	1	1007	1	1	1007	1007
1008	2020-04-09 12:30:00	2020-04-09 12:30:00	1008	1	1008	1	1	1008	1008
1009	2020-04-09 12:30:00	2020-04-09 12:30:00	1009	1	1009	1	1	1009	1009

b) Left outer join tickets and transactions based on mode of transition

MySQL Workbench

Query 1

```

1 * show databases;
2 * use mydb;
3 * select * from customers left join
4 transactions on
5 transactions.transaction_id and
6 transactions.transaction_amount > 1000;

```

customer_id	customer_name	customer_phone	customer_email	customer_address	customer_city	customer_state	customer_zip	customer_country	customer_created_at
1000	John Doe	1234567890	john.doe@example.com	123 Main St	New York	NY	10001	USA	2020-04-09 12:30:00
1001	Jane Smith	0987654321	jane.smith@example.com	456 Elm St	Los Angeles	CA	90001	USA	2020-04-09 12:30:00
1002	Mike Johnson	1122334455	mike.johnson@example.com	789 Oak St	Chicago	IL	60601	USA	2020-04-09 12:30:00
1003	Sarah Brown	2233445566	sarah.brown@example.com	101 Pine St	San Francisco	CA	94101	USA	2020-04-09 12:30:00
1004	David Wilson	3344556677	david.wilson@example.com	202 Cedar St	Seattle	WA	98101	USA	2020-04-09 12:30:00
1005	Emily Davis	4455667788	emily.davis@example.com	303 Birch St	Portland	OR	97201	USA	2020-04-09 12:30:00
1006	Chris Miller	5566778899	chris.miller@example.com	404 Maple St	Denver	CO	80201	USA	2020-04-09 12:30:00
1007	Alex Taylor	6677889900	alex.taylor@example.com	505 Spruce St	Phoenix	AZ	85001	USA	2020-04-09 12:30:00
1008	Olivia White	7788990011	olivia.white@example.com	606 Ash St	San Diego	CA	92101	USA	2020-04-09 12:30:00
1009	Noah Black	8899001122	noah.black@example.com	707 Hickory St	San Jose	CA	95101	USA	2020-04-09 12:30:00

c) Left outer join customers and transactions based on amount transitioned by joining 2 tables in between (customers_has_tickets and tickets)

3. RIGHT OUTER JOINS – 3 QUERIES

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

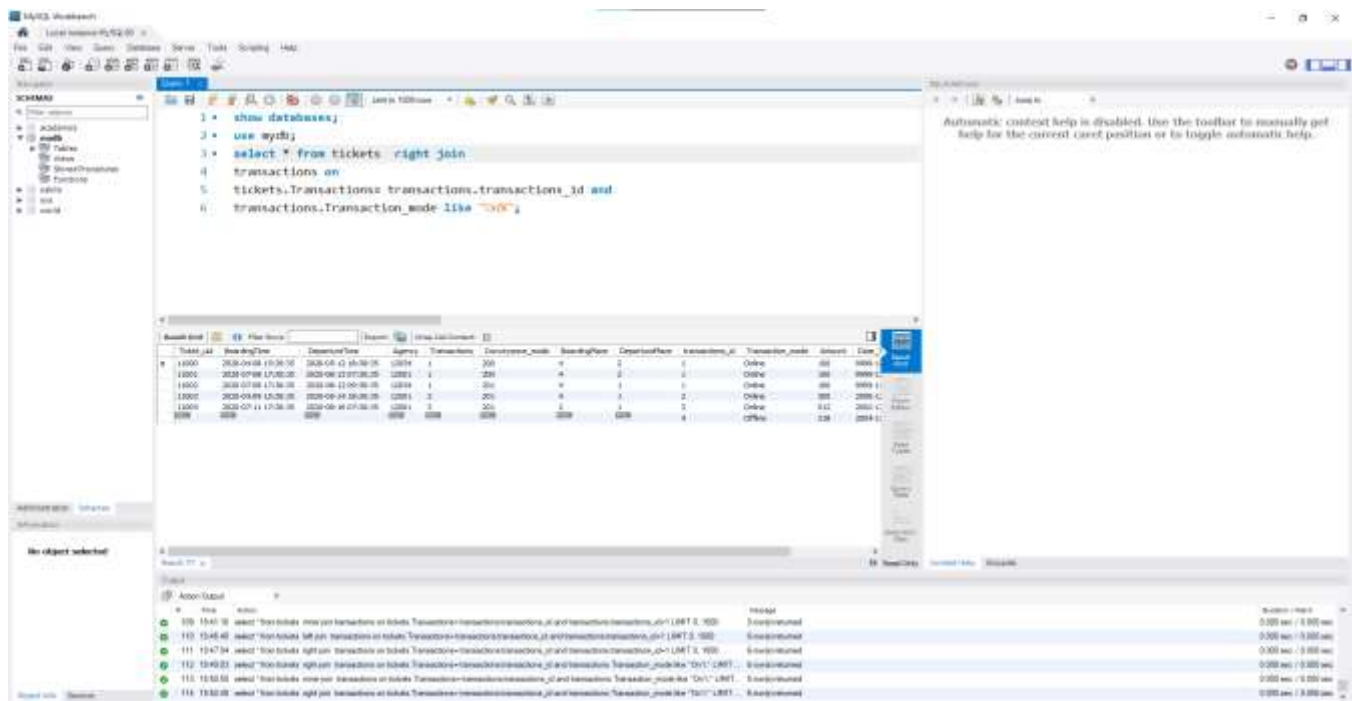
```
1. [Clicking the statement under the MySQL icon]
2. use mydb;
3. select * from tickets right join
4. transactions on
5. tickets.transactions= transactions.transactions_id and
6. transactions.transactions_id=i;
```

The query results are displayed in a table with the following columns: Ticket_id, BookingTime, DepartureTime, Agency, Transactions, Conversion_rate, BookingPlace, DeparturePlace, transactions_id, Transaction_rate, amount, Date. The results show 11 rows of data.

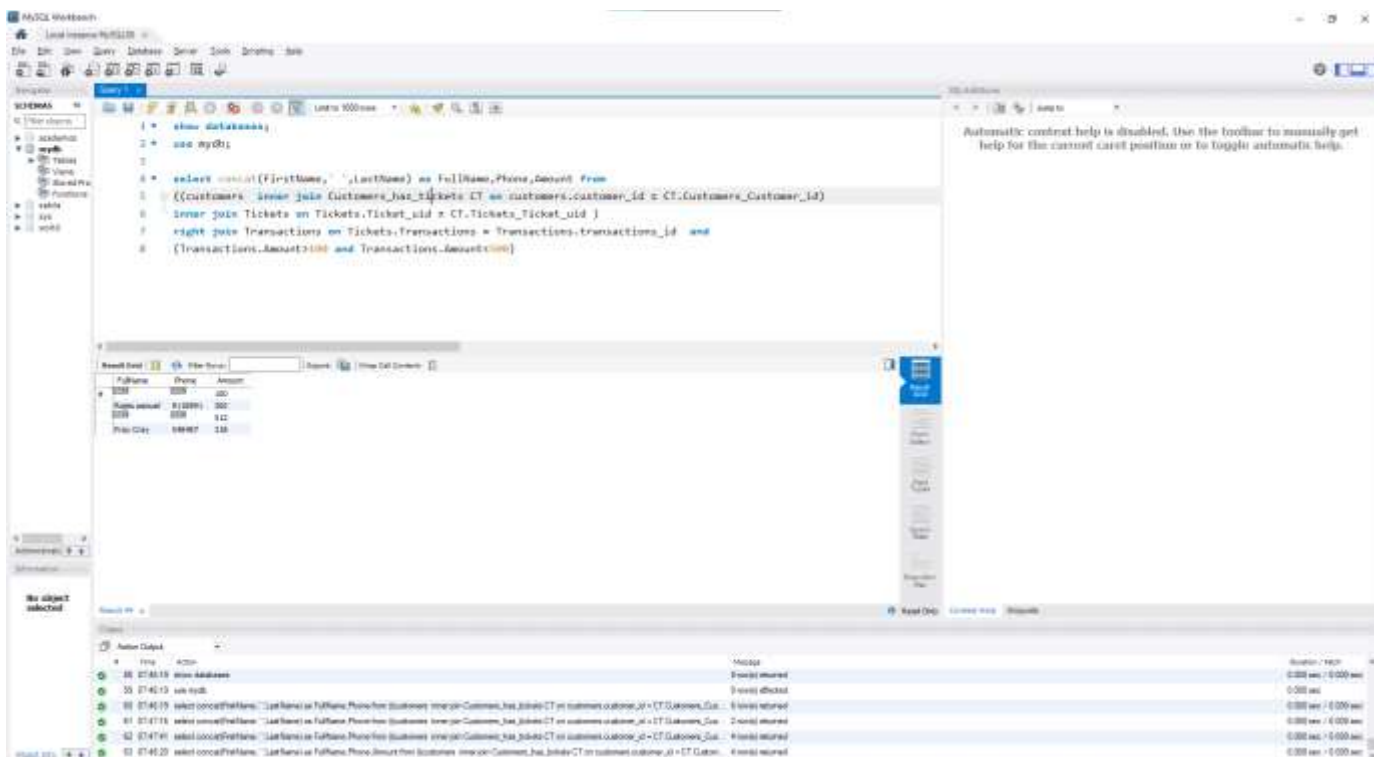
Below the query results, the 'Actions' tab is active, showing a list of actions performed by the user. The actions are as follows:

Time	Action	Message	Duration
10:07:05	select * from tickets on tickets.transactions=transactions_id and transactions_id=i LIMIT 0, 1000	Query returned	0.000 sec / 0.000 sec
10:07:06	select * from tickets on tickets.transactions=transactions_id and transactions_id=i LIMIT 0, 1000	Query returned	0.000 sec / 0.000 sec
10:07:07	select * from tickets on tickets.transactions=transactions_id and transactions_id=i LIMIT 0, 1000	Query returned	0.000 sec / 0.000 sec
10:07:08	select * from tickets on tickets.transactions=transactions_id and transactions_id=i LIMIT 0, 1000	Query returned	0.000 sec / 0.000 sec
10:07:09	select * from tickets on tickets.transactions=transactions_id and transactions_id=i LIMIT 0, 1000	Query returned	0.000 sec / 0.000 sec
10:07:10	select * from tickets on tickets.transactions=transactions_id and transactions_id=i LIMIT 0, 1000	Query returned	0.000 sec / 0.000 sec
10:07:11	select * from tickets on tickets.transactions=transactions_id and transactions_id=i LIMIT 0, 1000	Query returned	0.000 sec / 0.000 sec
10:07:12	select * from tickets on tickets.transactions=transactions_id and transactions_id=i LIMIT 0, 1000	Query returned	0.000 sec / 0.000 sec
10:07:13	select * from tickets on tickets.transactions=transactions_id and transactions_id=i LIMIT 0, 1000	Query returned	0.000 sec / 0.000 sec
10:07:14	select * from tickets on tickets.transactions=transactions_id and transactions_id=i LIMIT 0, 1000	Query returned	0.000 sec / 0.000 sec
10:07:15	select * from tickets on tickets.transactions=transactions_id and transactions_id=i LIMIT 0, 1000	Query returned	0.000 sec / 0.000 sec

a) Right outer join tickets and transactions based on transition id



b) Right outer join tickets and transactions based on mode of transition



c) Right outer join customers and transactions based on amount transitioned by joining 2 tables in between (customers_has_tickets and tickets)

