

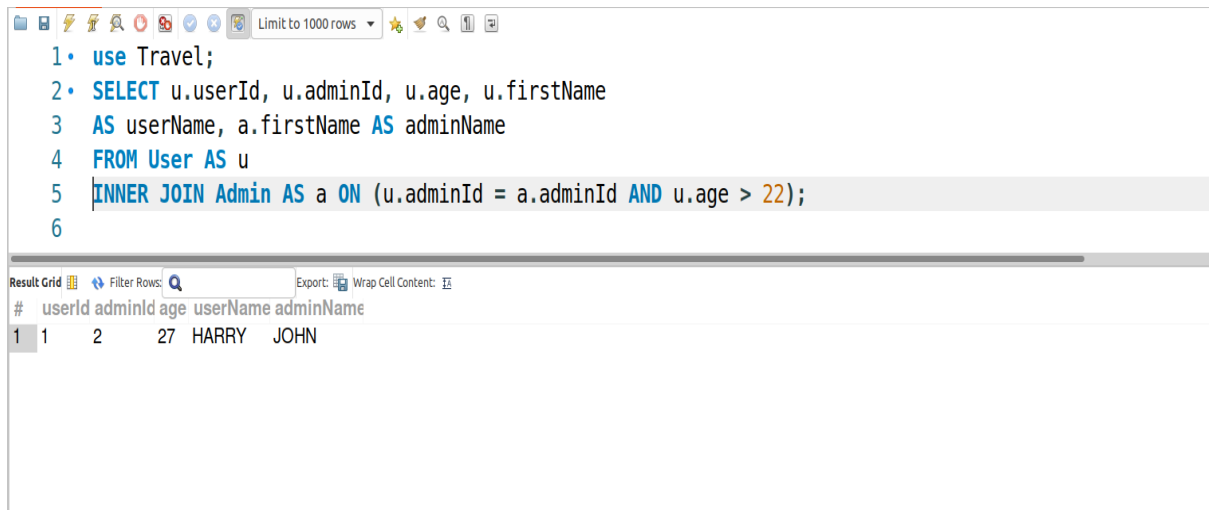
Assignment – 5

1. INNER JOIN

a.

Query:

```
SELECT u.userId, u.adminId, u.age, u.firstName  
AS userName, a.firstName AS adminName  
FROM User AS u  
INNER JOIN Admin AS a ON (u.adminId = a.adminId AND u.age > 22);
```



The screenshot shows a SQL query editor with a toolbar at the top containing icons for file operations, execution, and settings. The query text is as follows:

```
1 • use Travel;  
2 • SELECT u.userId, u.adminId, u.age, u.firstName  
3 AS userName, a.firstName AS adminName  
4 FROM User AS u  
5 INNER JOIN Admin AS a ON (u.adminId = a.adminId AND u.age > 22);  
6
```

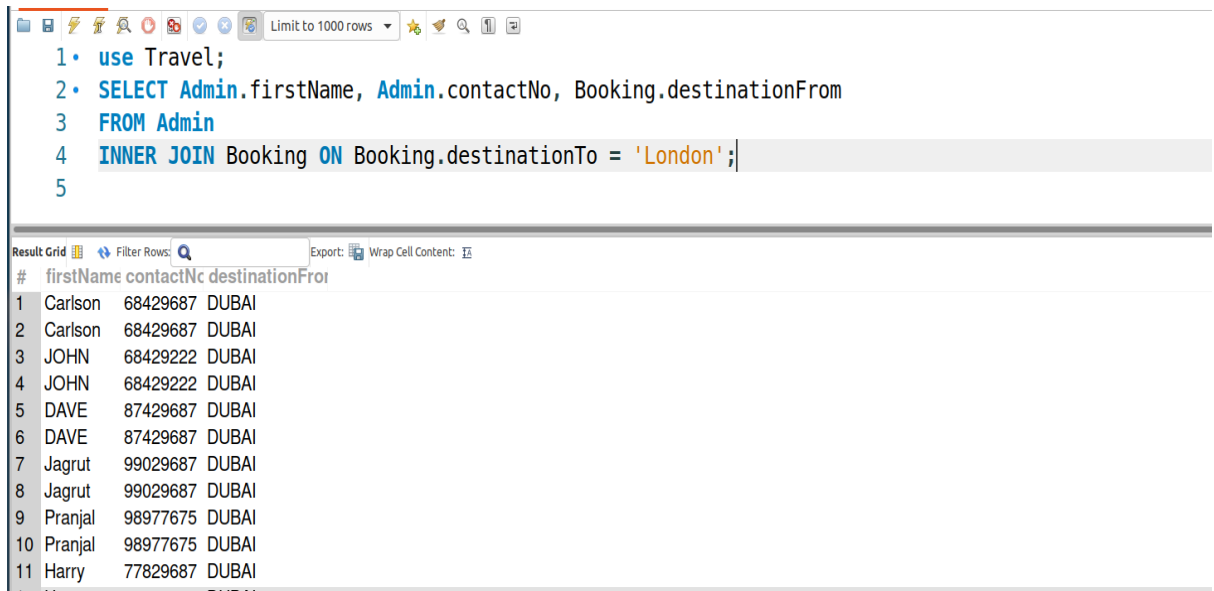
Below the query editor is a 'Result Grid' section. It includes a 'Filter Rows' search bar and an 'Export' button. The result table has the following structure and data:

#	userId	adminId	age	userName	adminName
1	1	2	27	HARRY	JOHN

b.

Query:

```
SELECT Admin.firstName, Admin.contactNo, Booking.destinationFrom
FROM Admin
INNER JOIN Booking ON Booking.destinationTo = 'London';
```



The screenshot shows a database query editor with a toolbar at the top containing icons for file operations, undo, redo, and a 'Limit to 1000 rows' dropdown. The query text is as follows:

```
1 • use Travel;
2 • SELECT Admin.firstName, Admin.contactNo, Booking.destinationFrom
3 FROM Admin
4 INNER JOIN Booking ON Booking.destinationTo = 'London';
5
```

Below the query editor is a 'Result Grid' with a search bar and an 'Export' button. The results are displayed in a table with the following columns: #, firstName, contactNo, and destinationFrom. The data rows are numbered 1 through 11.

#	firstName	contactNo	destinationFrom
1	Carlson	68429687	DUBAI
2	Carlson	68429687	DUBAI
3	JOHN	68429222	DUBAI
4	JOHN	68429222	DUBAI
5	DAVE	87429687	DUBAI
6	DAVE	87429687	DUBAI
7	Jagrut	99029687	DUBAI
8	Jagrut	99029687	DUBAI
9	Pranjal	98977675	DUBAI
10	Pranjal	98977675	DUBAI
11	Harry	77829687	DUBAI

C.

Query:

```
SELECT u.userId, u.age, a.adminId, b.bookingAgencyId, u.firstName
AS userName, a.firstName
AS adminName, b.name AS bookingAgencyName
FROM ((User AS u
INNER JOIN Admin AS a ON (u.adminId = a.adminId AND u.age < 23))
INNER JOIN BookingAgency AS b ON (b.adminId = a.adminId));
```

Limit to 1000 rows

```
1 • use Travel;
2
3 • SELECT u.userId, u.age, a.adminId, b.bookingAgencyId, u.firstName AS userName, a.firstName
4 AS adminName, b.name AS bookingAgencyName
5 FROM ((User AS u
6 INNER JOIN Admin AS a ON (u.adminId = a.adminId AND u.age < 23))
7 INNER JOIN BookingAgency AS b ON (b.adminId = a.adminId));
8
9
10
```

Result Grid Filter Rows: Export: Wrap Cell Content:

#	userId	age	adminId	bookingAgencyId	userName	adminName	bookingAgencyName
1	4	18	3	1	Sean	DAVE	READY GO
2	5	21	1	2	Kmann	Carlson	Fare Portal
3	3	22	2	3	Katy	JOHN	CLARK BOOKING
4	2	22	2	3	Ronald	JOHN	CLARK BOOKING
5	5	21	1	4	Kmann	Carlson	EASY Portal
6	4	18	3	5	Sean	DAVE	TRAVEL US

2. LEFT OUTER JOIN

a.

Query:

```
SELECT Advertisement.price
FROM Advertisement LEFT OUTER JOIN BookingAgency
ON Advertisement.bookingAgencyId =
BookingAgency.bookingAgencyId;
```

```
1 • use Travel;
2
3 • SELECT Advertisement.price
4 FROM Advertisement |
5 LEFT OUTER JOIN BookingAgency
6 ON Advertisement.bookingAgencyId = BookingAgency.bookingAgencyId;
7
8
```

#	price
1	5000
2	2000
3	500
4	3000
5	500

b.

Query:

```
SELECT u.userId, u.adminId, a.adminId, u.firstName AS userName,  
       u.age AS userAge,  
       a.firstName AS adminName, COUNT(u.userId) AS ageGroup  
FROM User AS u  
LEFT OUTER JOIN Admin AS a ON (u.adminId = 2)  
GROUP BY u.age  
HAVING ageGroup < 5;
```

MySQL Workbench

unconnected x Test x

File Edit View Query Database Server Tools Scripting Help

Travel Database Joins Assignment x

Limit to 1000 rows

```
29 • SELECT u.userId, u.adminId, a.adminId, u.firstName AS userName, u.age AS userAge,  
30     a.firstName AS adminName, COUNT(u.userId) AS ageGroup  
31     FROM User AS u  
32     LEFT OUTER JOIN Admin AS a ON (u.adminId = 2)  
33     GROUP BY u.age  
34     HAVING ageGroup < 5;
```

Result Grid Filter Rows: Export: Wrap Cell Content: F5

	userId	adminId	adminId	userName	userAge	adminName	ageGroup
▶ 3	1	NULL		John	23	NULL	1
4	1	NULL		James	21	NULL	1
5	2	1		Harry	27	Katy	2
6	2	1		John	24	Katy	4

C.

Query:

```
SELECT t.transportId, b.bookingAgencyId, t.transportType, t.charges
AS travelCharges, b.name
FROM transport AS t
LEFT OUTER JOIN BookingAgency AS b ON (t.bookingAgencyId=1 AND
t.transportType='Train')
ORDER BY t.charges;
```

MySQL Workbench

unconnected x Test x

File Edit View Query Database Server Tools Scripting Help

Travel Database Joins Assignment*

Limit to 1000 rows

```
35
36 • SELECT t.transportId, b.bookingAgencyId, t.transportType, t.charges AS travelCharges, b.name
37 FROM transport AS t
38 LEFT OUTER JOIN BookingAgency AS b ON (t.bookingAgencyId=1 AND t.transportType='Train')
39 ORDER BY t.charges;
40
```

Result Grid

	transportId	bookingAgencyId	transportType	travelCharges	name
4	NULL	Bus	500	NULL	
5	NULL	Bus	750	NULL	
3	1	Train	2500	Expedia Group	
3	2	Train	2500	Fare Portal	
1	NULL	Flight	5000	NULL	
2	NULL	Flight	10000	NULL	

3. RIGHT OUTER JOIN

a.

Query:

```
SELECT b.bookingAgencyId, a.advertisementId, b.name, a.price AS
AdvertisementPrice
FROM BookingAgency AS b
RIGHT OUTER JOIN Advertisement AS a ON (a.bookingAgencyId=2)
ORDER BY a.price;
```

MySQL Workbench

unconnected x Test x

File Edit View Query Database Server Tools Scripting Help

Travel Database Joins Assignment

Limit to 1000 rows

```
42
43 • SELECT b.bookingAgencyId, a.advertisementId, b.name, a.price AS AdvertisementPrice
44 FROM BookingAgency AS b
45 RIGHT OUTER JOIN Advertisement AS a ON (a.bookingAgencyId=2)
46 ORDER BY a.price;
47
```

<

Result Grid Filter Rows: Export: Wrap Cell Contents:

	transportId	bookingAgencyId	transportType	travelCharges	name
4	4	1	Bus	500	Expedia Group
5	5	1	Bus	750	Fare Portal
3	3	2	Train	2500	Expedia Group
3	3	2	Train	2500	Fare Portal
1	1	2	Flight	5000	Expedia Group
2	2	2	Flight	10000	Fare Portal

b.

Query:

```
SELEct u.userId, p.paymentId, u.firstName AS UserName,  
p.paymentType, p.paymentStatus, COUNT(p.paymentType) AS  
PaymentGroup  
FROM User AS u  
RIGHT OUTER JOIN Payment AS p ON (u.userId=p.userId)  
GROUP BY p.paymentType;
```

The screenshot shows the MySQL Workbench interface. The top toolbar includes icons for file operations, database management, and query execution. The main editor window displays the following SQL query:

```
47  
48 • SELEct u.userId, p.paymentId, u.firstName AS UserName, p.paymentType, p.paymentStatus, COUNT(p.paymentType) AS  
49 FROM User AS u  
50 RIGHT OUTER JOIN Payment AS p ON (u.userId=p.userId)  
51 GROUP BY p.paymentType;  
52
```

Below the query editor, the 'Result Grid' tab is active, showing the results of the query. The results are displayed in a table with the following columns: userId, paymentId, UserName, paymentType, paymentStatus, and PaymentGroup. The table contains three rows of data:

	userId	paymentId	UserName	paymentType	paymentStatus	PaymentGroup
▶ 3	1	John	UPI	Completed	1	
4	2	James	Paytm	Incomplete	1	
5	3	Harry	COD	Completed	1	

C.

Query:

```
SELECT b.bookingAgencyId, p.placeId, b.name AS
BookingAgencyName, p.name AS PlaceName, p.pincode,
COUNT(p.name) AS PlaceCount
FROM BookingAgency AS b
RIGHT OUTER JOIN Places AS p ON
(p.bookingAgencyId=b.bookingAgencyId)
GROUP BY p.name
HAVING PlaceCount > 1;
```

The screenshot shows the MySQL Workbench interface. The top toolbar includes icons for file operations, editing, and database management. The main window displays a SQL query in the 'Scripting' tab. The query is a RIGHT OUTER JOIN between the 'BookingAgency' and 'Places' tables, grouped by 'p.name' and filtered by 'PlaceCount > 1'. Below the query editor, the 'Result Grid' is visible, showing a single row of results. The columns are 'bookingAgencyId', 'placeId', 'BookingAgencyName', 'PlaceName', 'pincode', and 'PlaceCount'. The data row shows values: 2, 3, Fare Portal, Paris, 89123, and 2.

```
52
53 • SELECT b.bookingAgencyId, p.placeId, b.name AS BookingAgencyName, p.name AS PlaceName, p.pincode, COUNT(p.name)
54 FROM BookingAgency AS b
55 RIGHT OUTER JOIN Places AS p ON (p.bookingAgencyId=b.bookingAgencyId)
56 GROUP BY p.name
57 HAVING PlaceCount > 1;
```

bookingAgencyId	placeId	BookingAgencyName	PlaceName	pincode	PlaceCount
2	3	Fare Portal	Paris	89123	2