

Part A

Question 1

[1pt] List car rental companies which have a mileage of at least 27 miles/gallon.

```
select *
from bycar
where MILEAGE >= 27;
```

Script Output x Query Result x

SQL | All Rows Fetched: 3 in 0.015 seconds

	TID	RENTALCOMPANY	MILEAGE
1	16	Personal	30
2	20	Personal	29
3	21	Personal	27

Question 2

[1pt] List trip IDs taken on train costing strictly more than \$150.

Start Page x CSC452_Retake x

Worksheet Query Builder

```
select TID
from trips
where TRAVELMODE='Train' and Fare>150;
```

Query Result x

SQL | All Rows Fetched: 5 in 0.018 seconds

	TID
1	4
2	6
3	10
4	12
5	24

OR

Start Page x CSC452_Retake x

Worksheet Query Builder

```
select bytrain.TID
from bytrain, trips
where trips.tid = bytrain.TID and trips.FARE>150;
```

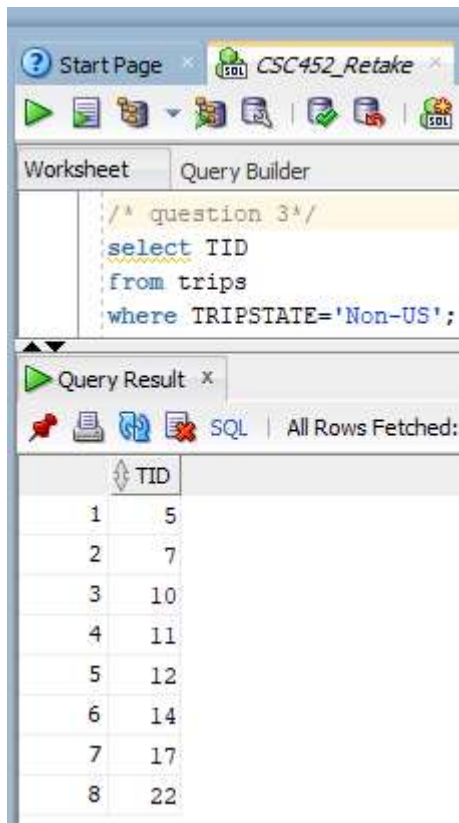
Query Result x

SQL | All Rows Fetched: 5 in 0.018 seconds

	TID
1	4
2	6
3	10
4	12
5	24

Question 3

1. [1pt] Find trip IDs and their fare that are not taken in the US i.e., `Non-US` trips.



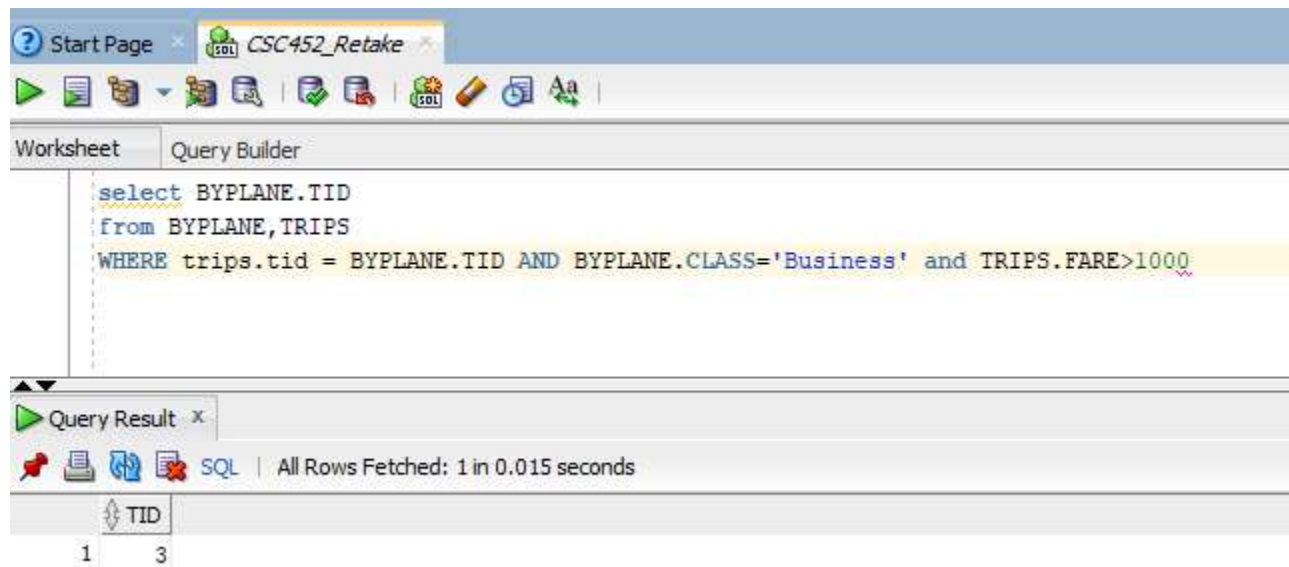
The screenshot shows a SQL query in the Query Builder window. The query is: `/* question 3 */
select TID
from trips
where TRIPSTATE='Non-US';`

The Query Result window shows the following data:

	TID
1	5
2	7
3	10
4	11
5	12
6	14
7	17
8	22

Question 4

[1pt] Find the business class plane trip IDs that are greater than \$1000.



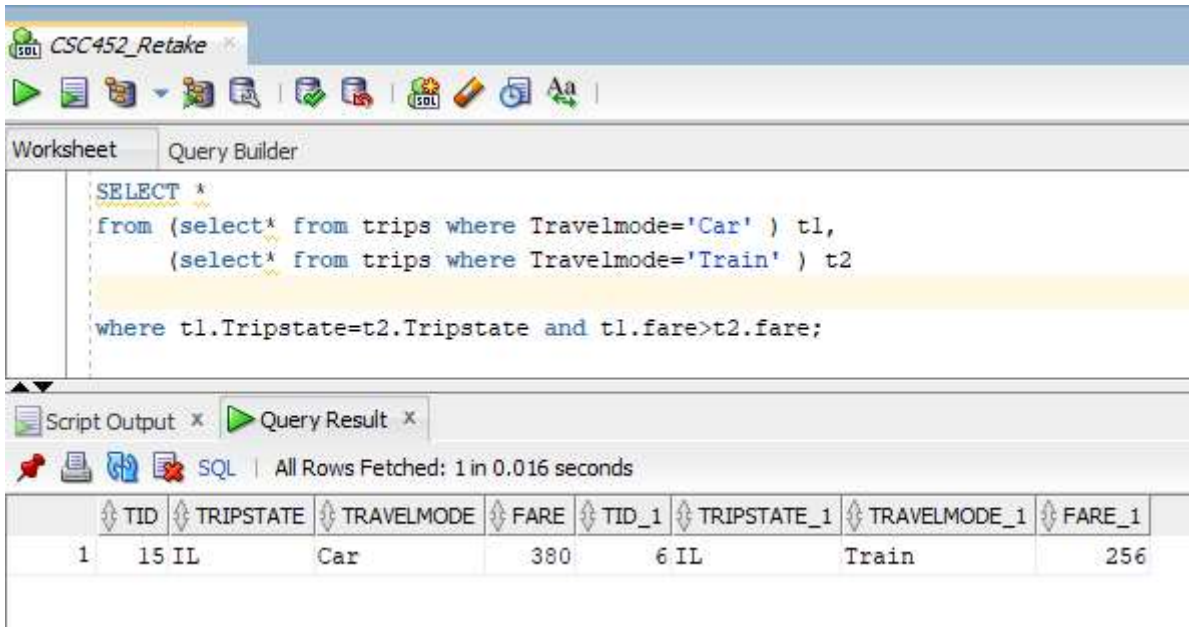
The screenshot shows a SQL query in the Query Builder window. The query is: `select BYPLANE.TID
from BYPLANE, TRIPS
WHERE trips.tid = BYPLANE.TID AND BYPLANE.CLASS='Business' and TRIPS.FARE>1000`

The Query Result window shows the following data:

	TID
1	3

Question 5

[2pt] Find any car trip more expensive than a trip taken on a train in the same state or outside the country.



The screenshot shows a SQL query builder interface with a query that finds car trips more expensive than train trips in the same state or outside the country. The query is as follows:

```
SELECT *
from (select* from trips where Travelmode='Car' ) t1,
     (select* from trips where Travelmode='Train' ) t2

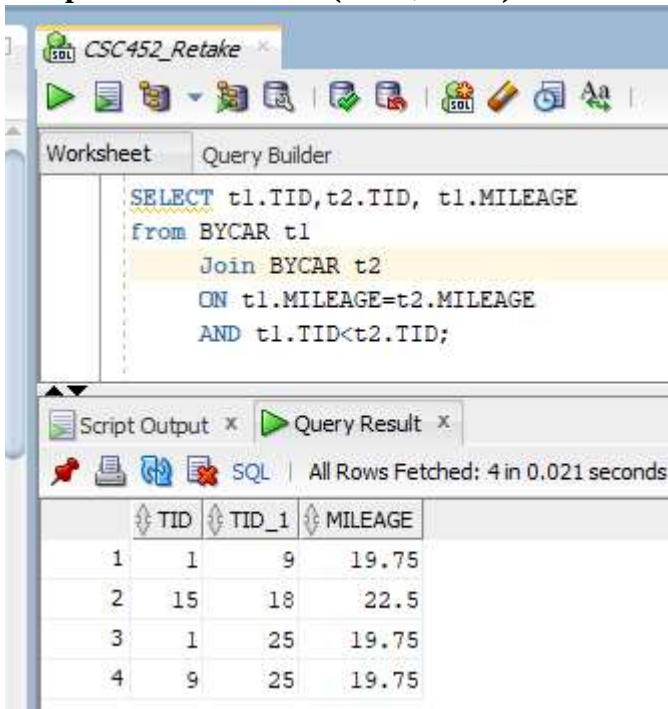
where t1.Tripstate=t2.Tripstate and t1.fare>t2.fare;
```

The query result shows one row where a car trip (TID 1, Tripstate IL, Fare 380) is more expensive than a train trip (TID 6, Tripstate IL, Fare 256).

TID	TRIPSTATE	TRAVELMODE	FARE	TID_1	TRIPSTATE_1	TRAVELMODE_1	FARE_1
1	15 IL	Car	380	6 IL		Train	256

Question 6

[2pt] List pairs of distinct trips that have exactly the same value of mileage. **Note a pair of distinct trips is of the format: (TID1, TID2). This distinct pair is not the same as the pair (TID2, TID1)**



The screenshot shows a SQL query builder interface with a query that finds pairs of distinct trips with the same mileage. The query is as follows:

```
SELECT t1.TID,t2.TID, t1.MILEAGE
from BYCAR t1
Join BYCAR t2
ON t1.MILEAGE=t2.MILEAGE
AND t1.TID<t2.TID;
```

The query result shows four rows of pairs of distinct trips with the same mileage.

TID	TID_1	MILEAGE
1	9	19.75
2	18	22.5
3	25	19.75
4	25	19.75

Question 7

[2pt] List pairs of distinct train trips that do not have the same speed.

CSC452_Retake

Worksheet Query Builder

```
select t1.TID, t1.TRAINSPEED, t2.TID, t2.TRAINSPEED
from bytrain t1
Join bytrain t2
On t1.TRAINSPEED!=t2.TRAINSPEED
AND t1.TID<t2.TID;
```

Script Output x Query Result x

SQL | All Rows Fetched: 26 in 0.025 seconds

	TID	TRAINSPEED	TID_1	TRAINSPEED_1
1	4	50	6	30
2	4	50	10	70
3	4	50	12	25
4	4	50	14	70
5	4	50	22	5
6	4	50	24	20
7	6	30	8	50
8	6	30	10	70
9	6	30	12	25
10	6	30	14	70
11	6	30	22	5
12	6	30	24	20
13	8	50	10	70
14	8	50	12	25
15	8	50	14	70
16	8	50	22	5
17	8	50	24	20
18	10	70	12	25
19	10	70	22	5
20	10	70	24	20
21	12	25	14	70
22	12	25	22	5
23	12	25	24	20
24	14	70	22	5
25	14	70	24	20
26	22	5	24	20

Question 8

[2pt] Find those pair of trips in the same state with the same mode of travel. List such pairs only once. In other words, given a pair (TID1,TID2) do NOT list (TID2,TID1).

The screenshot shows a SQL query in a 'Query Builder' window. The query is as follows:

```
select t1.TID, t1.TRAVELMODE, t1.TRIPSTATE, t2.TID
from trips t1
Join trips t2
On t1.TRAVELMODE=t2.TRAVELMODE
And t1.TRIPSTATE=t2.TRIPSTATE
AND t1.TID<t2.TID;
```

Below the query, the 'Query Result' window displays 26 rows of data. The columns are TID, TRAVELMODE, TRIPSTATE, and TID_1. The data is as follows:

TID	TRAVELMODE	TRIPSTATE	TID_1
1	5 Plane	Non-US	7
2	5 Plane	Non-US	11
3	7 Plane	Non-US	11
4	10 Train	Non-US	12
5	10 Train	Non-US	14
6	12 Train	Non-US	14
7	1 Car	IL	15
8	1 Car	IL	16
9	15 Car	IL	16
10	5 Plane	Non-US	17
11	7 Plane	Non-US	17
12	11 Plane	Non-US	17
13	9 Car	IN	18
14	3 Plane	MD	19
15	1 Car	IL	20
16	15 Car	IL	20
17	16 Car	IL	20
18	1 Car	IL	21
19	15 Car	IL	21
20	16 Car	IL	21
21	20 Car	IL	21
22	10 Train	Non-US	22
23	12 Train	Non-US	22
24	14 Train	Non-US	22
25	13 Plane	IL	23
26	4 Train	MD	24

Question 9

[4pt] Find a state in which trips have been taken by all three modes of transportation: train, plane, and car.

The screenshot shows a SQL query builder interface with the following query:

```
select trips.TID, trips.Tripstate, trips.TRAVELMODE
from trips trips, bycar c, bytrain t, byplane p
where trips.TID=c.tid and trips.tid=c.tid and trips.tid=t.tid and trips.tid=p.tid;
```

The query result pane shows the following columns: TID, TRIPSTATE, TRAVELMODE.

Question 10

[4pt] Find the details of a) the most costly trip, b) the cheapest trip, taken by either the car, train, or plane. Write two separate queries for (a) and (b). **Write the last query as a self-join with basic SQL operators (Filter, Project, Rename, Join (cross-join, natural join), Union, Intersect, and Difference). Do not use ALL, ANY, DISTINCT, GROUP BY, HAVING, MAX, MIN, ORDER BY.**

a)

The screenshot shows a SQL query builder interface with the following query:

```
select * from trips where fare not in
(select a.fare from trips a, trips b where a.fare<p.fare);
```

The query result pane shows the following columns: TID, TRIPSTATE, TRAVELMODE, FARE. The result is as follows:

TID	TRIPSTATE	TRAVELMODE	FARE
1	7 Non-US	Plane	5000

b)

The screenshot shows a SQL query builder interface with the following query:

```
select * from trips where fare not in
(select a.fare from trips a, trips b where a.fare>b.fare);
```

The query result pane shows the following columns: TID, TRIPSTATE, TRAVELMODE, FARE. The result is as follows:

TID	TRIPSTATE	TRAVELMODE	FARE
1	8 NM	Train	13