

Lavinia Wang

#1473704

HW 1

The screenshot shows a database management system interface with the following details:

- Connections:** ComputerStore.sql, csc453, C.sql
- Reports:** All Reports, Analytic View Reports, Data Dictionary Reports, Data Modeler Reports, OLAP Reports, TimesTen Reports, User Defined Reports
- Worksheet - Query Builder:** Contains three numbered queries:
 1. List car rental companies which have a mileage of at least 27 miles/gallon.

```
1 --Student Name: Lavinia Wong
2 --Student ID: 1473704
3
4 SELECT * FROM trips;
5 SELECT * FROM bycar;
6 SELECT * FROM bytrain;
7 SELECT * FROM byplane;
8
9
10 1. List car rental companies which have a mileage of at least 27 miles/gallon.
```
 2. List trip IDs taken on train costing strictly more than \$150.

```
11 SELECT rentalcompany
12 FROM bycar
13 WHERE mileage >= 27;
```
 3. Find trip IDs and their fare that are not taken in the US i.e., 'Non-US' trips.

```
14
15 2. List trip IDs taken on train costing strictly more than $150.
16 SELECT tid
17 FROM trips
18 WHERE (travelmode = 'Train') AND fare > 150;
19
20 3. Find trip IDs and their fare that are not taken in the US i.e., 'Non-US' trips.
```
- Query Result:** Shows the result of query 2:

RENTALCOMPANY
1 Personal
2 Personal
3 Personal
- SQL History:** Shows the executed SQL statement and its execution details:

SQL	Connect...	TimeStamp	Type	Executed	Duration...
SELECT rentalcompany FROM bycar WHERE mileage >= 27;	csc453	01-OCT-18 04.47 PM	SQL	1	0.015

The screenshot shows a database management system interface with the following details:

- Connections:** ComputerStore.sql, csc453, C.sql
- Reports:** All Reports, Analytic View Reports, Data Dictionary Reports, Data Modeler Reports, OLAP Reports, TimesTen Reports, User Defined Reports
- Worksheet - Query Builder:** Contains three numbered queries:
 1. List car rental companies which have a mileage of at least 27 miles/gallon.

```
1 --Student Name: Lavinia Wong
2 --Student ID: 1473704
3
4 SELECT * FROM trips;
5 SELECT * FROM bycar;
6 SELECT * FROM bytrain;
7 SELECT * FROM byplane;
8
9
10 1. List car rental companies which have a mileage of at least 27 miles/gallon.
```
 2. List trip IDs taken on train costing strictly more than \$150.

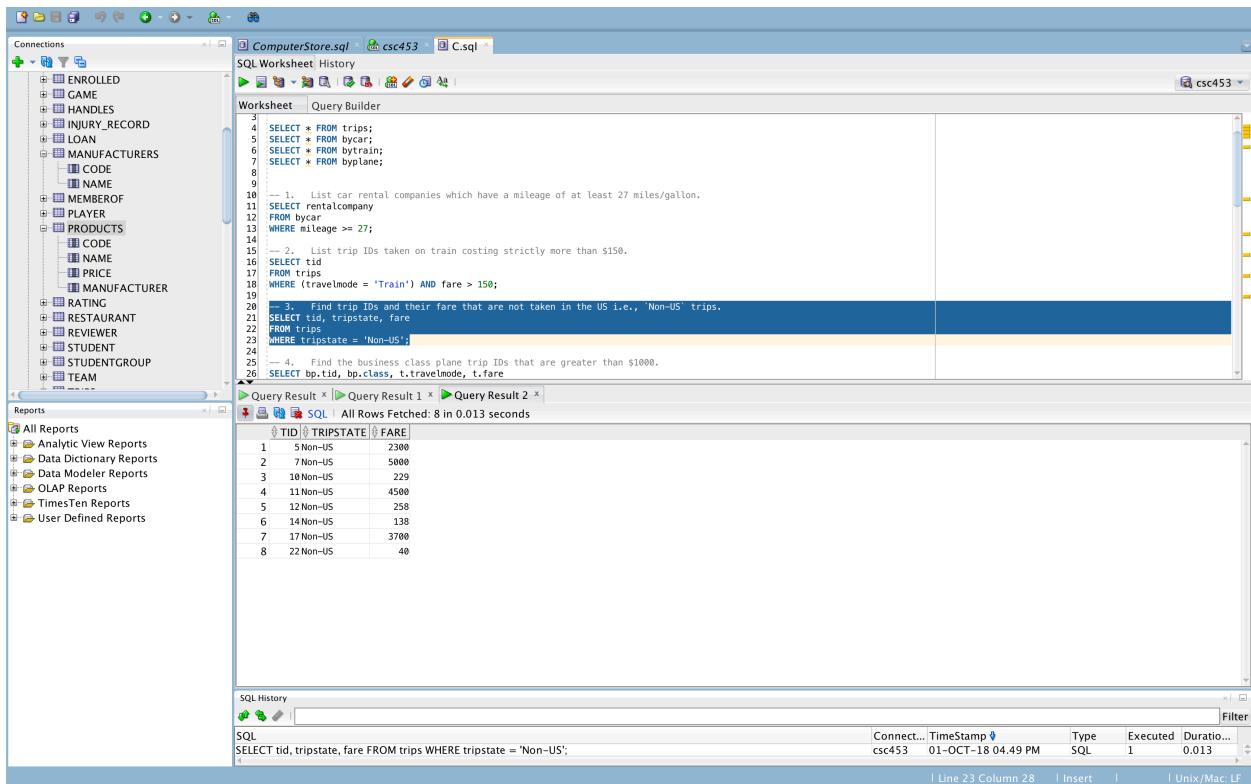
```
11 SELECT tid
12 FROM trips
13 WHERE (travelmode = 'Train') AND fare > 150;
14
15 2. List trip IDs taken on train costing strictly more than $150.
```
 3. Find trip IDs and their fare that are not taken in the US i.e., 'Non-US' trips.

```
16 SELECT tid, tripstate, fare
17 FROM trips
18 WHERE tripstate = 'Non-US';
19
20 3. Find trip IDs and their fare that are not taken in the US i.e., 'Non-US' trips.
```
- Query Result:** Shows the result of query 2:

TID
1 4
2 6
3 10
4 12
5 24
- SQL History:** Shows the executed SQL statement and its execution details:

SQL	Connect...	TimeStamp	Type	Executed	Duration...
SELECT tid FROM trips WHERE (travelmode = 'Train') AND fare > 150;	csc453	01-OCT-18 04.49 PM	SQL	1	0.02

Lavinia Wang
 #1473704
 HW 1



ComputerStore.sql - csc453 - C.sql

```

3
4   SELECT * FROM trips;
5   SELECT * FROM bycar;
6   SELECT * FROM bytrain;
7   SELECT * FROM byplane;
8
9
10  1. List car rental companies which have a mileage of at least 27 miles/gallon.
11  SELECT rentalcompany
12  FROM bycar
13  WHERE mileage >= 27;
14
15  2. List trip IDs taken on train costing strictly more than $150.
16  SELECT tid
17  FROM trips
18  WHERE (travelmode = 'Train') AND fare > 150;
19
20  3. Find trip IDs and their fare that are not taken in the US i.e., 'Non-US' trips.
21  SELECT tid, tripstate, fare
22  FROM trips
23  WHERE tripstate = 'Non-US';
24
25  4. Find the business class plane trip IDs that are greater than $1000.
26  SELECT bp.tid, bp.class, t.travelmode, t.fare

```

Query Result x | Query Result 1 x | Query Result 2 x

TID	TRIPSTATE	FARE
1	Non-US	2300
2	Non-US	5000
3	Non-US	229
4	Non-US	4500
5	Non-US	258
6	Non-US	138
7	Non-US	3700
8	Non-US	40

SQL History

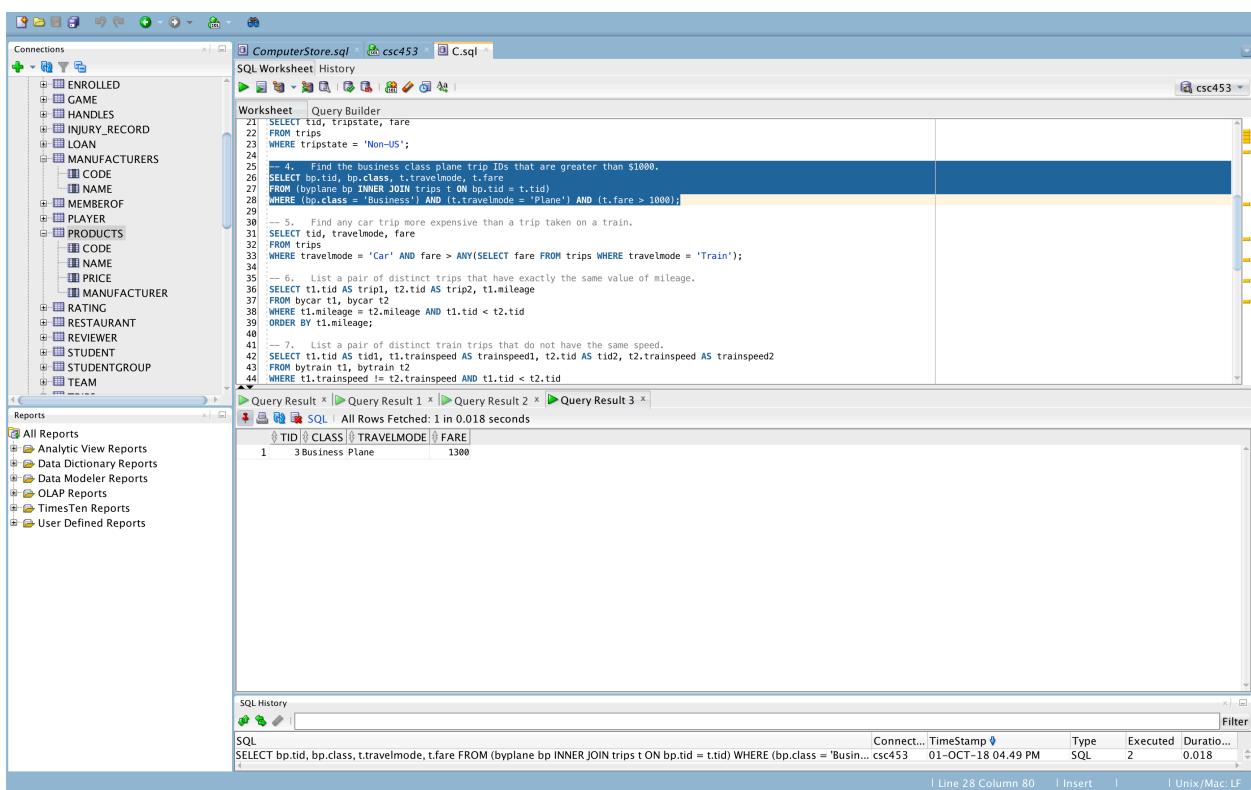
```

SQL
SELECT tid, tripstate, fare FROM trips WHERE tripstate = 'Non-US';

```

Connect... TimeStamp Type Executed Duration...
csc453 01-OCT-18 04:49 PM SQL 1 0.013

| Line 23 Column 28 | Insert | Unix/Mac: LF



ComputerStore.sql - csc453 - C.sql

```

21  SELECT tid, tripstate, fare
22  FROM trips
23  WHERE tripstate = 'Non-US';
24
25  4. Find the business class plane trip IDs that are greater than $1000.
26  SELECT bp.tid, bp.class, t.travelmode, t.fare
27  FROM (byplane bp INNER JOIN trips t ON bp.tid = t.tid)
28  WHERE (bp.class = 'Business') AND (t.travelmode = 'Plane') AND (t.fare > 1000);
29
30  5. Find any car trip more expensive than a trip taken on a train.
31  SELECT tid, travelmode, fare
32  FROM trips
33  WHERE travelmode = 'Car' AND fare > ANY(SELECT fare FROM trips WHERE travelmode = 'Train');
34
35  6. List a pair of distinct trips that have exactly the same value of mileage.
36  SELECT t1.tid AS trip1, t2.tid AS trip2, t1.mileage
37  FROM bycar t1, bycar t2
38  WHERE t1.mileage = t2.mileage AND t1.tid < t2.tid
39  ORDER BY t1.mileage;
40
41  7. List a pair of distinct train trips that do not have the same speed.
42  SELECT t1.tid AS tid1, t1.trainspeed AS trainspeed1, t2.tid AS tid2, t2.trainspeed AS trainspeed2
43  FROM bytrain t1, bytrain t2
44  WHERE t1.trainspeed != t2.trainspeed AND t1.tid < t2.tid

```

Query Result x | Query Result 1 x | Query Result 2 x | Query Result 3 x

TID	CLASS	TRAVELMODE	FARE
1	Business	Plane	1300

SQL History

```

SQL
SELECT bp.tid, bp.class, t.travelmode, t.fare FROM (byplane bp INNER JOIN trips t ON bp.tid = t.tid) WHERE (bp.class = 'Business') AND (t.travelmode = 'Plane') AND (t.fare > 1000);

```

Connect... TimeStamp Type Executed Duration...
csc453 01-OCT-18 04:49 PM SQL 2 0.018

| Line 28 Column 80 | Insert | Unix/Mac: LF

Lavinia Wang

#1473704

HW 1

SQL Worksheet History

```
21 SELECT tid, tripstate, fare
22 FROM trips
23 WHERE tripstate = 'Non-US';
24
25 -- 4. Find the business class plane trip IDs that are greater than $1000.
26 SELECT bp.tid, bp.class, t.travelmode, t.fare
27 FROM (byplane bp INNER JOIN trips t ON bp.tid = t.tid)
28 WHERE (bp.class = 'Business') AND (t.travelmode = 'Plane') AND (t.fare > 1000);
29
30 -- 5. Find any car trip more expensive than a trip taken on a train.
31 SELECT tid, travelmode, fare
32 FROM trips
33 WHERE travelmode = 'Car' AND fare > ANY(SELECT fare FROM trips WHERE travelmode = 'Train');
34
35 -- 6. List a pair of distinct trips that have exactly the same value of mileage.
36 SELECT t1.tid AS trip1, t2.tid AS trip2, t1.mileage
37 FROM bycar t1, bycar t2
38 WHERE t1.mileage = t2.mileage AND t1.tid < t2.tid
39 ORDER BY t1.mileage;
40
41 -- 7. List a pair of distinct train trips that do not have the same speed.
42 SELECT t1.tid AS tid1, t1.trainspeed AS trainspeed1, t2.tid AS tid2, t2.trainspeed AS trainspeed2
43 FROM bytrain t1, bytrain t2
44 WHERE t1.trainspeed != t2.trainspeed AND t1.tid < t2.tid
```

Query Result x | Query Result 1 x | Query Result 2 x | Query Result 3 x | Query Result 4 x

TID	TRAVELMODE	FARE
1	9 Car	400
2	15 Car	380
3	21 Car	156
4	25 Car	114
5	1 Car	100
6	18 Car	88
7	20 Car	75
8	16 Car	59

All Reports

SQL History

Connect...	TimeStamp	Type	Executed	Duration...
csc453	01-OCT-18 04:49 PM	SQL	2	0.014

SQL Worksheet History

```
21 SELECT tid, tripstate, fare
22 FROM trips
23 WHERE tripstate = 'Non-US';
24
25 -- 4. Find the business class plane trip IDs that are greater than $1000.
26 SELECT bp.tid, bp.class, t.travelmode, t.fare
27 FROM (byplane bp INNER JOIN trips t ON bp.tid = t.tid)
28 WHERE (bp.class = 'Business') AND (t.travelmode = 'Plane') AND (t.fare > 1000);
29
30 -- 5. Find any car trip more expensive than a trip taken on a train.
31 SELECT tid, travelmode, fare
32 FROM trips
33 WHERE travelmode = 'Car' AND fare > ANY(SELECT fare FROM trips WHERE travelmode = 'Train');
34
35 -- 6. List a pair of distinct trips that have exactly the same value of mileage.
36 SELECT t1.tid AS trip1, t2.tid AS trip2, t1.mileage
37 FROM bycar t1, bycar t2
38 WHERE t1.mileage = t2.mileage AND t1.tid < t2.tid
39 ORDER BY t1.mileage;
40
41 -- 7. List a pair of distinct train trips that do not have the same speed.
42 SELECT t1.tid AS tid1, t1.trainspeed AS trainspeed1, t2.tid AS tid2, t2.trainspeed AS trainspeed2
43 FROM bytrain t1, bytrain t2
44 WHERE t1.trainspeed != t2.trainspeed AND t1.tid < t2.tid
```

Query Result x | Query Result 1 x | Query Result 2 x | Query Result 3 x | Query Result 4 x | Query Result 5 x

TRIP1	TRIP2	MILEAGE
1	1	9
2	9	25
3	1	25
4	15	18

All Reports

SQL History

Connect...	TimeStamp	Type	Executed	Duration...
csc453	01-OCT-18 04:49 PM	SQL	1	0.011

Lavinia Wang

#1473704

HW 1

SQL Worksheet History

```
40
41  -- 7. List a pair of distinct train trips that do not have the same speed.
42  SELECT t1.tripid AS trip1, t2.tripid AS trip2, t1.trainspeed AS trainspeed1,
43    t2.trainspeed AS trainspeed2
44  FROM bytrain t1, bytrain t2
45  WHERE t1.trainspeed != t2.trainspeed AND t1.tripid < t2.tripid
46
```

Worksheet Query Builder

Query Result 1 x | Query Result 2 x | Query Result 3 x | Query Result 4 x | Query Result 5 x | Query Result 6 x

All Rows Fetched: 26 in 0.017 seconds

TID1	TRAINSPED1	TID2	TRAINSPED2
1	12.25	14.70	
2	12.25	24.20	
3	12.25	22.5	
4	6.30	8.50	
5	6.30	10.70	
6	6.30	22.5	
7	6.30	14.70	
8	6.30	12.25	
9	6.30	24.20	
10	22.5	24.20	
11	8.50	24.20	
12	8.50	22.5	
13	8.50	14.70	
14	8.50	12.25	
15	8.50	10.70	
16	4.50	24.20	
17	4.50	22.5	
18	4.50	14.70	
19	4.50	12.25	
20	4.50	6.30	
21	4.50	10.70	
22	10.70	22.5	
23	10.70	12.25	
24	14.70	22.5	
25	14.70	24.20	
26	10.70	24.20	

SQL History

Connect...	TimeStamp	Type	Executed	Duration...
csc453	01-OCT-18 04.49 PM	SQL	2	0.017

Line 45 Column 24 | Insert | Unix/Mac: LF

SQL Worksheet History

```
46
47  -- 8. Find those pair of trips in the same state with the same mode of travel. List such pairs only once.
48  SELECT t1.tripid AS trip1, t2.tripid AS trip2, t1.tripstate, t1.travelmode
49  FROM trips t1, trips t2
50  WHERE t1.tripstate = t2.tripstate AND t1.tripstate NOT IN ('Non-US') AND t1.travelmode = t2.travelmode
51  ORDER BY t1.tripstate;
52
```

Worksheet Query Builder

Query Result 1 x | Query Result 2 x | Query Result 3 x | Query Result 4 x | Query Result 5 x | Query Result 6 x | Query Result 7 x

All Rows Fetched: 14 in 0.012 seconds

TRIP1	TRIP2	TRIPSTATE	TRAVELMODE
1	1	15 IL	Car
2	1	16 IL	Car
3	15	16 IL	Car
4	15	20 IL	Car
5	13	23 IL	Plane
6	20	21 IL	Car
7	1	20 IL	Car
8	16	20 IL	Car
9	1	21 IL	Car
10	15	21 IL	Car
11	16	21 IL	Car
12	9	18 IN	Car
13	4	24 MD	Train
14	3	19 MD	Plane

SQL History

Connect...	TimeStamp	Type	Executed	Duration...
csc453	01-OCT-18 04.50 PM	SQL	1	0.012

Line 51 Column 23 | Insert | Unix/Mac: LF

Lavinia Wang

#1473704

HW 1

The screenshot shows the SQL Worksheet History window in a database management system. The connection is set to 'csc453'. The worksheet contains two queries:

```
52 — 9. Find a state in which trips have been taken by all three modes of transportation: train, plane, and car.
53 (SELECT tripstate
54 FROM trips
55 WHERE travelmode = 'Car')
56 INTERSECT
57 (SELECT tripstate
58 FROM trips
59 WHERE travelmode = 'Train')
60 INTERSECT
61 (SELECT tripstate
62 FROM trips
63 WHERE travelmode = 'Plane');
64
65 — 10. Find the details of a) the most costly trip, b) the cheapest trip taken by either the air, rail, or car. Write two separate queries.
```

The results of the first query are displayed in a grid:

TRIPSTATE	TID
IL	1

The SQL History panel at the bottom shows the executed SQL statement and its details:

SQL	Connect...	TimeStamp	Type	Executed	Duration...
(SELECT tripstate FROM trips WHERE travelmode = 'Car') INTERSECT (SELECT tripstate FROM trips WHERE travelmode = 'Train') INTERSECT (SELECT tripstate FROM trips WHERE travelmode = 'Plane');	csc453	01-OCT-18 04.50 PM	SQL	3	0.022

The screenshot shows the SQL Worksheet History window in a database management system. The connection is set to 'csc453'. The worksheet contains two queries:

```
62 — 10. Find the details of a) the most costly trip, b) the cheapest trip taken by either the air, rail, or car. Write two separate queries.
63 (SELECT * FROM trips
64 WHERE travelmode = 'Plane');
65
66 — 10. Find the details of a) the most costly trip, b) the cheapest trip taken by either the air, rail, or car. Write two separate queries.
67 SELECT tid, fare
68 FROM trips
69 WHERE fare = (SELECT MAX(fare) from trips);
70
71 SELECT tid, fare
72 FROM trips
73 WHERE fare = (SELECT MIN(fare) from trips)
74
75
76
```

The results of the second query are displayed in a grid:

TID	FARE
1	7
1	5000

The SQL History panel at the bottom shows the executed SQL statement and its details:

SQL	Connect...	TimeStamp	Type	Executed	Duration...
SELECT tid, fare FROM trips WHERE fare = (SELECT MAX(fare) from trips);	csc453	01-OCT-18 04.57 PM	SQL	2	0.017

Lavinia Wang
#1473704
HW 1

The screenshot shows the Oracle SQL Developer interface. On the left, the Connections tree shows a connection to 'csc453'. The central area is a 'Worksheet' titled 'Query Builder' containing the following SQL code:

```
01 --> /*@csc453*/
02   SELECT tid, fare
03     FROM trips
04    WHERE travelmode = 'Plane';
05
06  --> 10. Find the details of a) the most costly trip, b) the cheapest trip taken by either the air, rail, or car. Write two separate queries.
07
08  SELECT tid, fare
09    FROM trips
10   WHERE fare = (SELECT MAX(fare) from trips);
11
12  SELECT tid, fare
13    FROM trips
14   WHERE fare = (SELECT MIN(fare) from trips);
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
```

Below the worksheet is a 'Query Result' tab showing the output of the first query:

TID	FARE
1	8
13	

The bottom right corner of the interface shows the status bar with 'All Rows Fetched: 1 in 0.008 seconds'.

At the bottom, the 'SQL History' pane displays the executed SQL statement:

```
SQL
SELECT tid, fare FROM trips WHERE fare = (SELECT MIN(fare) from trips)
```

The history table includes columns: Connect..., TimeStamp, Type, Executed, Duration, Line, Column, Insert, and Unix/Mac:LF. The entry for the query is:

Connect...	TimeStamp	Type	Executed	Duration...	Line	Column	Insert	Unix/Mac:LF
csc453	01-OCT-18 04.57 PM	SQL	2	0.008	4			