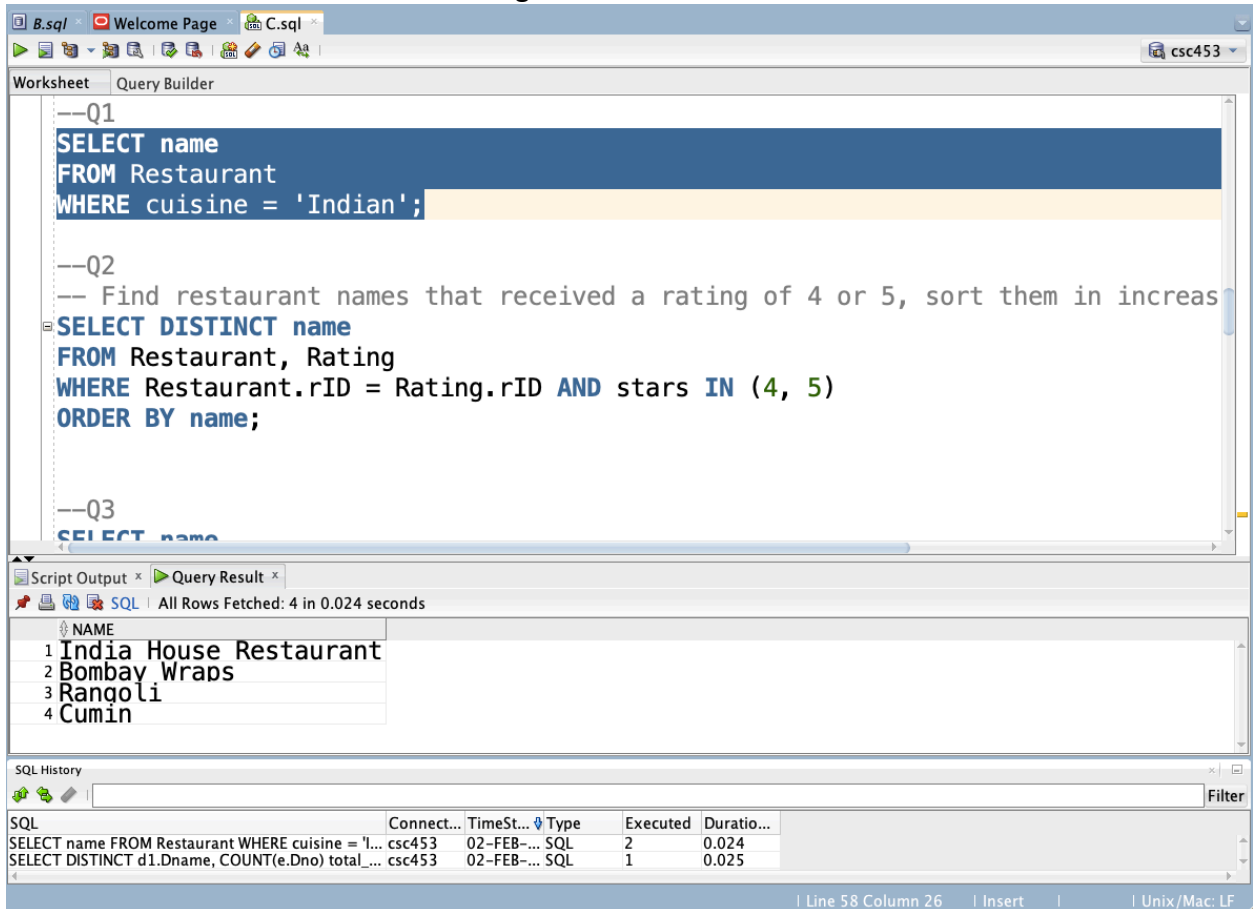


Ximan Liu
CSC 453
HW2
Part C Screenshots

1. Find the name of all restaurants offering Indian cuisine



The screenshot displays a SQL IDE interface with three queries defined:

```
--Q1
SELECT name
FROM Restaurant
WHERE cuisine = 'Indian';

--Q2
-- Find restaurant names that received a rating of 4 or 5, sort them in increasing order
SELECT DISTINCT name
FROM Restaurant, Rating
WHERE Restaurant.rID = Rating.rID AND stars IN (4, 5)
ORDER BY name;

--Q3
SELECT name
```

The results pane shows the output for the first query:

	NAME
1	India House Restaurant
2	Bombay Wraps
3	Rangoli
4	Cumin

The SQL History pane at the bottom shows the execution of the first two queries:

SQL	Connect...	TimeSt...	Type	Executed	Duration...
SELECT name FROM Restaurant WHERE cuisine = 'I...	csc453	02-FEB-...	SQL	2	0.024
SELECT DISTINCT d1.Dname, COUNT(e.Dno) total...	csc453	02-FEB-...	SQL	1	0.025

2. Find restaurant names that received a rating of 4 or 5, sort them in increasing order.

The screenshot shows a SQL IDE with two queries and their results. The first query, labeled --Q2, selects distinct restaurant names with a rating of 4 or 5. The second query, labeled --Q3, selects restaurant names that do not have a rating. The results of the first query are displayed in a table below the script.

```
--Q2
SELECT DISTINCT name
FROM Restaurant, Rating
WHERE Restaurant.rID = Rating.rID AND stars IN (4, 5)
ORDER BY name;

--Q3
SELECT name
FROM Restaurant
WHERE rID NOT IN (SELECT rID FROM Rating);
```

Script Output x Query Result x

SQL | All Rows Fetched: 4 in 0.014 seconds

	NAME
1	India House Restaurant
2	Jade Court
3	MindHin Cuisine
4	Shanghai Terrace

SQL History

SQL	Connect...	TimeSt...	Type	Executed	Duration...
SELECT DISTINCT name FROM Restaurant, Rating ...	csc453	02-FEB-...	SQL	5	0.014
SELECT name FROM Restaurant, Rating WHERE Res...	csc453	02-FEB-...	SQL	1	0.019
SELECT name FROM Restaurant WHERE cuisine = 'I...	csc453	02-FEB-...	SQL	2	0.024
SELECT DISTINCT d1.Dname, COUNT(e.Dno) total ...	csc453	02-FEB-...	SQL	1	0.025

| Line 65 Column 15 | Insert | Modified | Unix/Mac: LF

- Find the names of all restaurants that have no rating.

The screenshot shows a SQL IDE with two queries and their results.

Query Q3:

```
SELECT name
FROM Restaurant
WHERE rID NOT IN (SELECT rID FROM Rating);
```

Query Q4:

```
SELECT name
FROM Reviewer
INNER JOIN Rating USING(vID)
WHERE ratingDate IS NULL;
```

Query Results:

Script Output x Query Result x
 All Rows Fetched: 2 in 0.015 seconds

	NAME
1	Shanghai Inn
2	Bombay Wraps

SQL History:

SQL	Connect...	TimeSt...	Type	Executed	Duration...
SELECT name FROM Restaurant WHERE rID NOT IN ...	csc453	02-FEB-...	SQL	3	0.015
SELECT DISTINCT name FROM Restaurant, Rating ...	csc453	02-FEB-...	SQL	5	0.014
SELECT name FROM Restaurant, Rating WHERE Res...	csc453	02-FEB-...	SQL	1	0.019
SELECT name FROM Restaurant WHERE cuisine = 'I...	csc453	02-FEB-...	SQL	2	0.024

Line 71 Column 43 | Insert | Modified | Unix/Mac: LF

- Some reviewers didn't provide a date with their rating. Find the names of all reviewers who have ratings with a NULL value for the date.

The screenshot shows a SQL IDE with two queries and their results. The first query, Q3, is a simple SELECT statement. The second query, Q4, is a more complex JOIN query. The results of Q4 are displayed in a table with two rows.

```

--Q3
SELECT name
FROM Restaurant
WHERE rID NOT IN (SELECT rID FROM Rating);

--Q4
SELECT name
FROM Reviewer
INNER JOIN Rating USING(vID)
WHERE ratingDate IS NULL;

```

Script Output x Query Result x

SQL | All Rows Fetched: 2 in 0.019 seconds

	NAME
1	Daniel L.
2	Suikev S.

SQL History

SQL	Connect...	TimeSt...	Type	Executed	Duration...
SELECT name FROM Reviewer INNER JOIN Rating U...	csc453	02-FEB-...	SQL	4	0.019
SELECT name FROM Restaurant WHERE rID NOT IN ...	csc453	02-FEB-...	SQL	3	0.015
SELECT DISTINCT name FROM Restaurant, Rating ...	csc453	02-FEB-...	SQL	5	0.014
SELECT name FROM Restaurant. Rating WHERE Res...	csc453	02-FEB-...	SQL	1	0.019

Line 78 Column 26 | Insert | Modified | Unix/Mac: LF

- For all cases where the same reviewer rated the same restaurant twice and gave it a higher rating the second time, return the reviewer's name and the name of the restaurant.

The screenshot shows a SQL IDE with two queries and their results.

Query Q5:

```
--Q5
SELECT Reviewer.name, Restaurant.name
FROM Restaurant
INNER JOIN Rating R1 USING(rID)
INNER JOIN Rating R2 USING(vID, rID)
INNER JOIN Reviewer USING(vID)
WHERE R1.ratingDate < R2.ratingDate AND R1.stars < R2.stars;
```

Query Q6:

```
--Q6
SELECT name, MAX(stars)
FROM Restaurant
INNER JOIN Rating USING(rID)
```

Script Output: All Rows Fetched: 1 in 0.034 seconds

NAME	NAME_1
1	Sarah M. India House Restaurant

SQL History:

SQL	Connect...	TimeSt...	Type	Executed	Duration...
SELECT Reviewer.name, Restaurant.name FROM R...	csc453	02-FEB-...	SQL	2	0.034
SELECT name FROM Reviewer INNER JOIN Rating U...	csc453	02-FEB-...	SQL	4	0.019
SELECT name FROM Restaurant WHERE rID NOT IN ...	csc453	02-FEB-...	SQL	3	0.015
SELECT DISTINCT name FROM Restaurant. Rating ...	csc453	02-FEB-...	SQL	5	0.014

Line 87 Column 61 | Insert | Modified | Unix/Mac: LF

- For each restaurant that has at least one rating, find the highest number of stars that a restaurant received. Return the restaurant name and number of stars. Sort by restaurant name.

The screenshot shows a SQL IDE interface with two queries and their results.

Query Q6:

```
--Q6
SELECT name, MAX(stars)
FROM Restaurant
INNER JOIN Rating USING(rID)
GROUP BY rID, name
ORDER BY name;
```

Query Q7:

```
--Q7
SELECT name, (MAX(stars) - MIN(stars)) AS rating_spread
FROM Restaurant
INNER JOIN Rating USING(rID)
```

Query Results:

Script Output x Query Result x

SQL All Rows Fetched: 6 in 0.014 seconds

	NAME	MAX(STARS)
1	Cumin	3
2	India House Restaurant	4
3	Jade Court	4
4	MindHin Cuisine	5
5	Randoli	3
6	Shanghai Terrace	5

SQL History:

SQL	Connect...	TimeSt...	Type	Executed	Duration...
SELECT name, MAX(stars) FROM Restaurant INNER ...	csc453	02-FEB-...	SQL	3	0.014
SELECT Reviewer.name, Restaurant.name FROM R...	csc453	02-FEB-...	SQL	2	0.034
SELECT name FROM Reviewer INNER JOIN Rating U...	csc453	02-FEB-...	SQL	4	0.019
SELECT name FROM Restaurant WHERE rID NOT IN ...	csc453	02-FEB-...	SQL	3	0.015

Line 95 Column 15 | Insert | Modified | Unix/Mac: LF

- For each restaurant, return the name and the 'rating spread', that is, the difference between highest and lowest ratings given to that restaurant. Sort by rating spread from highest to lowest, then by restaurant name.

The screenshot shows a SQL IDE with two queries and their results. The top panel displays the SQL code for two queries, Q7 and Q8. The bottom panel shows the results of these queries.

Query Q7:

```

--Q7
SELECT name, (MAX(stars) - MIN(stars)) AS rating_spread
FROM Restaurant
INNER JOIN Rating USING(rID)
GROUP BY rID, name
ORDER BY rating_spread DESC, name;

```

Query Q8:

```

--Q8
SELECT AVG(Indian.avg) - AVG(Chinese.avg)
FROM
(

```

Query Results:

The results of Query Q7 are shown in a table with two columns: NAME and RATING_SPREAD.

NAME	RATING_SPREAD
1 India House Restaurant	2
2 Jade Court	2
3 Shanghai Terrace	2
4 Cumin	1
5 MinHui Cuisine	1
6 Rangoli	1

The bottom panel shows the SQL History, which includes the following queries:

SQL	Connect...	TimeSt...	Type	Executed	Duration...
SELECT name, (MAX(stars) - MIN(stars)) AS rating...	csc453	02-FEB-...	SQL	4	0.015
SELECT name, MAX(stars) FROM Restaurant INNER ...	csc453	02-FEB-...	SQL	3	0.014
SELECT Reviewer.name, Restaurant.name FROM R...	csc453	02-FEB-...	SQL	2	0.034
SELECT name FROM Reviewer INNER JOIN Rating U...	csc453	02-FEB-...	SQL	4	0.019

- Find the difference between the average rating of Indian restaurants and the average rating of Chinese restaurants. (Make sure to calculate the average rating for each restaurant, then the average of those averages for Indian and Chinese restaurants. Don't just calculate the overall average rating for Indian and Chinese restaurants.) Note: The difference can be negative.

