

DB-LAB-TASK-3

Roll No: 24P-0706

Name: Aazan Noor Khuwaja

Dept: BS-CS

Section : 4B

1. Display all developer records:

Command:

```
SELECT * FROM dev_survey
```

Output:

The screenshot shows the DB Browser for SQLite interface. The title bar reads "DB Browser for SQLite - /home/aazan-noor-khuwaja/Aazan_Data/4th_Semester/DB_Lab/lab2/tech_survey.db". The main window has a toolbar with File, Edit, View, Tools, Help, New Database, Open Database, Write Changes, Revert Changes, Open Project, Save Project, Attach Database, and Close Database. Below the toolbar is a menu bar with Database Structure, Browse Data, Edit Pragmas, and Execute SQL. The Execute SQL tab is selected. A SQL editor window titled "SQL 1" contains the query: "SELECT * FROM dev_survey". To the right of the SQL editor is a "Edit Database Cell" panel with "Mode: Text" set to "Text". The result of the query is displayed in a table with columns: dev_id, country, experience_years, primary_language, secondary_language, database_used, framework, and annual_salary. The table contains 21 rows of developer data. At the bottom of the interface, there are tabs for SQL Log, Plot, DB Schema, and Remote, along with a "UTF-8" encoding indicator.

dev_id	country	experience_years	primary_language	secondary_language	database_used	framework	annual_salary
1	Pakistan	2	Python	JavaScript	MySQL	Django	8800
2	USA	10	JavaScript	TypeScript	PostgreSQL	React	90000
3	Germany	7	C#	SQL	SQL Server	.NET	70000
4	UK	12	Python	R	PostgreSQL	Flask	85000
5	Canada	4	JavaScript	Python	MongoDB	Vue	60000
6	Pakistan	3	Java	Kotlin	MySQL	Spring	13200.0
7	USA	15	C++	Python	SQLite	Qt	110000
8	India	6	PHP	JavaScript	MySQL	Laravel	15000
9	France	8	Python	SQL	PostgreSQL	Django	65000
10	Australia	9	JavaScript	NULL	MongoDB	React	75000
11	Japan	11	Java	Python	Oracle	Spring	82000
12	Netherlands	6	Go	Python	PostgreSQL	NULL	68000
13	Pakistan	5	C++	Python	MySQL	NULL	15400.0
14	Sweden	10	Python	Julia	SQLite	Django	90000
15	India	2	JavaScript	HTML	Firebase	Angular	10000
16	South Africa	7	Python	JavaScript	PostgreSQL	FastAPI	55000
17	Pakistan	19	c++	java	sqlite3	FastAPI	146300
18	Vietnam	NULL	PHP	NULL	NULL	NULL	117000
19	India	23	c++	java	sqlite3	FastAPI	131000
20	Pakistan	19	c++	java	sqlite3	FastAPI	146300

2. Display only country, primary_language, annual_salary:

Command:

```
SELECT country,primary_language,annual_salary FROM dev_survey;
```

Output:

The screenshot shows the DB Browser for SQLite interface. The SQL tab contains the following query:

```
SELECT country,primary_language,annual_salary FROM dev_survey
```

The results table displays 21 rows of developer survey data:

	country	primary_language	annual_salary
1	Pakistan	Python	8800
2	USA	JavaScript	90000
3	Germany	C#	70000
4	UK	Python	85000
5	Canada	JavaScript	60000
6	Pakistan	Java	132000
7	USA	C++	110000
8	India	PHP	15000
9	America	Python	65000
10	Australia	JavaScript	75000
11	Japan	Java	82000
12	Netherlands	Go	68000
13	Pakistan	C++	154000
14	Sweden	Python	90000
15	India	JavaScript	10000
16	South Africa	Python	55000
17	Pakistan	C++	146300
18	Vietnam	NULL	117000
19	India	C++	131000
20	Pakistan	C++	146300

Execution finished without errors. Result: 21 rows returned in 0ms.

3. List developers earning more than 50,000 USD:

Command:

```
SELECT * FROM dev_survey
```

```
WHERE (annual_salary>50000);
```

Output:

The screenshot shows the DB Browser for SQLite interface. The SQL tab contains the following query:

```
SELECT * FROM dev_survey
WHERE (annual_salary>50000)
```

The results table displays 16 rows of developer survey data from countries where annual salary is greater than 50,000 USD:

	dev_id	country	experience_years	primary_language	secondary_language	database_used	framework	annual_salary
1	3	USA	10	JavaScript	TypeScript	PostgreSQL	React	90000
2	4	Germany	7	C#	SQL	SQL Server	.NET	70000
3	6	UK	12	Python	R	PostgreSQL	Flask	85000
4	7	Canada	4	JavaScript	Python	MongoDB	Vue	60000
5	9	USA	15	C++	Python	SQLite	Qt	110000
6	11	France	8	Python	SQL	PostgreSQL	Django	65000
7	12	Australia	9	JavaScript	NULL	MongoDB	React	75000
8	13	Japan	11	Java	Python	Oracle	Spring	82000
9	15	Netherlands	6	Go	Python	PostgreSQL	NULL	68000
10	18	Sweden	10	Python	Julia	SQLite	Django	90000
11	20	South Africa	7	Python	JavaScript	PostgreSQL	FastAPI	55000
12	21	Pakistan	19	C++	java	sqlite3	FastAPI	146300
13	22	Vietnam	NULL	NULL	PHP	NULL	NULL	117000
14	24	India	23	C++	java	sqlite3	FastAPI	131000
15	25	Pakistan	19	C++	java	sqlite3	FastAPI	146300
16	26	Bangladesh	7	C++	java	sqlite3	FastAPI	132000

Execution finished without errors. Result: 16 rows returned in 0ms.

4. Show developers from Pakistan earning less than 20,000 USD:

Command:

```
SELECT * FROM dev_survey  
WHERE country='Pakistan' AND annual_salary<20000;
```

Output:

The screenshot shows the DB Browser for SQLite interface. The title bar reads "DB Browser for SQLite - /home/aazan-noor-khuwaja/Aazan_Data/4th_Semester/DB_Lab/lab2/tech_survey.db". The toolbar includes "File", "Edit", "View", "Tools", and "Help". Below the toolbar are buttons for "New Database", "Open Database", "Write Changes", "Revert Changes", "Open Project", "Save Project", "Attach Database", and "Close Database". The menu bar has "Database Structure", "Browse Data", "Edit Pragmas", and "Execute SQL". The "Execute SQL" tab is selected. In the SQL editor, the following query is written:

```
1 SELECT * FROM dev_survey  
2 WHERE country='Pakistan' AND annual_salary<20000;
```

Below the SQL editor, the results are displayed in a table:

dev_id	country	experience_years	primary_language	secondary_language	database_used	framework	annual_salary	
1	Pakistan	2	Python	JavaScript	MySQL	Django	8800	
2	8	Pakistan	3	Java	Kotlin	MySQL	Spring	13200.0
3	17	Pakistan	5	C++	Python	MySQL	NULL	15400.0

At the bottom of the results window, the message "Execution finished without errors. Result: 3 rows returned in 6ms At line 1: SELECT * FROM dev_survey WHERE country='Pakistan' AND annual_salary<20000;" is shown.

5. Show developers who use Python and PostgreSQL:

Command:

```
SELECT *  
FROM dev_survey  
WHERE primary_language = 'Python' AND database_used = 'PostgreSQL';
```

Output:

6. Show developers from USA or Germany earning above 60,000 USD:

Command:

SELECT *

FROM dev_survey

WHERE country = 'USA' OR country = 'Germany' AND annual_salary>60000;

Output:

7. Show developers who do not use JavaScript:

Command:

```
SELECT *  
FROM dev_survey  
WHERE primary_language != 'JavaScript';
```

Output:

The screenshot shows the DB Browser for SQLite interface. The SQL tab contains the query: `SELECT * FROM dev_survey WHERE primary_language != 'JavaScript';`. The results pane displays a table with 16 rows of developer survey data. The columns are: dev_id, country, experience_years, primary_language, secondary_language, database_used, framework, and annual_salary. The data includes various programming languages like Python, C#, Java, etc., and frameworks like Django, Flask, Spring, etc. The annual salary ranges from 65000 to 132000. The results pane also shows a note at the bottom: "Execution finished without errors. Result: 16 rows returned in 5ms".

dev_id	country	experience_years	primary_language	secondary_language	database_used	framework	annual_salary
1	Pakistan	2	Python	JavaScript	MySQL	Django	8800
2	Germany	7	C#	SQL	SQL Server	.NET	70000
3	UK	12	Python	R	PostgreSQL	Flask	85000
4	Pakistan	3	Java	Kotlin	MySQL	Spring	132000.0
5	USA	15	C++	Python	SQLite	Qt	110000
6	India	6	PHP	JavaScript	MySQL	Laravel	15000
7	France	8	Python	SQL	PostgreSQL	Django	65000
8	Japan	11	Java	Python	Oracle	Spring	82000
9	Netherlands	6	Go	Python	PostgreSQL	NULL	68000
10	Pakistan	5	C++	Python	MySQL	NULL	15400.0
11	Sweden	10	Python	Julia	SQLite	Django	90000
12	South Africa	7	Python	JavaScript	PostgreSQL	FastAPI	55000
13	Pakistan	19	c++	java	sqlite3	FastAPI	NULL
14	India	23	c++	java	sqlite3	FastAPI	131000
15	Pakistan	19	c++	java	sqlite3	FastAPI	146300
16	Bangladesh	7	c++	java	sqlite3	FastAPI	132000

8. Show developers with experience between 5 and 10 years:

Command:

```
SELECT *  
FROM dev_survey  
WHERE experience_years BETWEEN 5 AND 10;
```

Output:

The screenshot shows the DB Browser for SQLite interface. The SQL tab contains the following query:

```

1 SELECT *
2 FROM dev_survey
3 WHERE experience_years BETWEEN 5 AND 10;

```

The results table displays 10 rows of developer survey data. The columns are: dev_id, country, experience_years, primary_language, secondary_language, database_used, framework, and annual_salary. The data includes various programming languages like JavaScript, C#, Python, PHP, Java, C++, Go, and others, along with their respective frameworks and salaries.

dev_id	country	experience_years	primary_language	secondary_language	database_used	framework	annual_salary
1	USA	10	JavaScript	TypeScript	PostgreSQL	React	90000
2	Germany	7	C#	SQL	SQL Server	.NET	70000
3	India	6	PHP	JavaScript	MySQL	Laravel	15000
4	France	8	Python	SQL	PostgreSQL	Django	65000
5	Australia	9	JavaScript	NULL	MongoDB	React	75000
6	Netherlands	6	Go	Python	PostgreSQL	NULL	68000
7	Pakistan	5	C++	Python	MySQL	NULL	15400.0
8	Sweden	10	Python	Julia	SQLite	Django	90000
9	South Africa	7	Python	JavaScript	PostgreSQL	FastAPI	55000
10	Bangladesh	7	c++	java	sqlite3	FastAPI	132000

Execution finished without errors.
Result: 10 rows returned in 5ms
At Line 1:
SELECT *

9. Show developers whose primary language is either one of Python, Java, C++,PHP:

Command:

SELECT *

FROM dev_survey

WHERE primary_language IN ('Python','Java','C++','PHP');

Output:

The screenshot shows the DB Browser for SQLite interface. The SQL tab contains the following query:

```

1 SELECT *
2 FROM dev_survey
3 WHERE primary_language IN ('Python','Java','C++','PHP');

```

The results table displays 10 rows of developer survey data. The columns are: dev_id, country, experience_years, primary_language, secondary_language, database_used, framework, and annual_salary. The data includes developers using Python, Java, C++, and PHP as their primary language, along with their respective frameworks and salaries.

dev_id	country	experience_years	primary_language	secondary_language	database_used	framework	annual_salary
1	Pakistan	2	Python	JavaScript	MySQL	Django	8800
2	UK	12	Python	R	PostgreSQL	Flask	85000
3	Pakistan	3	Java	Kotlin	MySQL	Spring	13200.0
4	USA	15	C++	Python	SQLite	Qt	110000
5	India	6	PHP	JavaScript	MySQL	Laravel	15000
6	France	8	Python	SQL	PostgreSQL	Django	65000
7	Japan	11	Java	Python	Oracle	Spring	82000
8	Pakistan	5	C++	Python	MySQL	NULL	15400.0
9	Sweden	10	Python	Julia	SQLite	Django	90000
10	South Africa	7	Python	JavaScript	PostgreSQL	FastAPI	55000

Execution finished without errors.
Result: 10 rows returned in 5ms
At Line 1:
SELECT *

10. Show developers whose framework starts with S:

Command:

```
SELECT *  
FROM dev_survey
```

```
WHERE framework LIKE 'S%';
```

Output:

The screenshot shows the DB Browser for SQLite interface. The SQL tab contains the following query:

```
1 SELECT *  
2 FROM dev_survey  
3 WHERE framework LIKE 'S%';
```

The results pane displays the following data:

dev_id	country	experience_years	primary_language	secondary_language	database_used	framework	annual_salary
1	Pakistan	8	Java	Kotlin	MySQL	Spring	13200.0
2	Japan	13	Java	Python	Oracle	Spring	82000

The status bar at the bottom indicates "Execution finished without errors. Result: 2 rows returned in 4ms".

11. Show developers whose database contains the word SQL:

Command:

```
SELECT *  
FROM dev_survey
```

```
WHERE database_used LIKE '%SQL%';
```

Output:

DB Browser for SQLite - /home/aazan-noor-khuwaja/Aazan_Data/4th_Semester/DB_Lab/lab2/tech_survey.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1

```
1 SELECT *
2 FROM dev_survey
3 WHERE database_used LIKE '%SQL%';
```

dev_id	country	experience_years	primary_language	secondary_language	database_used	framework	annual_salary
1	Pakistan	2	Python	JavaScript	MySQL	Django	8800
2	USA	10	JavaScript	TypeScript	PostgreSQL	React	90000
3	Germany	7	C#	SQL	SQL Server	.NET	70000
4	UK	12	Python	R	PostgreSQL	Flask	85000
5	Pakistan	3	Java	Kotlin	MySQL	Spring	13200.0
6	USA	15	C++	Python	SQLite	Qt	110000
7	India	6	PHP	JavaScript	MySQL	Laravel	15000
8	France	8	Python	SQL	PostgreSQL	Django	65000
9	Netherlands	6	Go	Python	PostgreSQL	NULL	68000
10	Pakistan	5	C++	Python	MySQL	NULL	15400.0
11	Sweden	10	Python	Julia	SQLite	Django	90000
12	South Africa	7	Python	JavaScript	PostgreSQL	FastAPI	55000
13	Pakistan	19	c++	java	sqlite3	FastAPI	NULL
14	INDIA	23	c++	java	sqlite3	FastAPI	131000
15	Pakistan	19	c++	java	sqlite3	FastAPI	146300
16	Bangladesh	7	c++	java	sqlite3	FastAPI	132000

Execution finished without errors.
Result: 16 rows returned in 5ms
At line 1:
SELECT *

Mode: Text

Type of data currently in cell: NULL

0 byte(s)

Apply

Remote

Select an identity to connect

DBHub.io Local Current Database

Name

SQL Log Plot DB Schema Remote

UTF-8

12. Show the top 5 highest-paid developers:

Command:

```
SELECT *
```

```
FROM dev_survey
```

```
ORDER BY annual_salary DESC
```

```
LIMIT 5;
```

Output:

The screenshot shows the DB Browser for SQLite interface. In the SQL tab, the following query is run:

```

1 SELECT *
2 FROM dev_survey
3 ORDER BY annual_salary DESC
4 LIMIT 5;

```

The results are displayed in a table:

dev_id	country	experience_years	primary_language	secondary_language	database_used	framework	annual_salary
1	Pakistan	19	C++	java	sqlite3	FastApi	146300
2	Bangladesh	7	C++	java	sqlite3	FastApi	132000
3	India	23	C++	java	sqlite3	FastApi	131000
4	Vietnam	22	NULL	PHP	NULL	NULL	117000
5	USA	9	C++	Python	SQLite	Qt	110000

Execution finished without errors. Result: 5 rows returned in 4ms At Line 1: SELECT *;

13. Show the lowest-paid developers sorted by salary:

Command:

SELECT *

FROM dev_survey

ORDER BY annual_salary ASC;

Output:

The screenshot shows the DB Browser for SQLite interface. In the SQL tab, the following query is run:

```

1 SELECT *
2 FROM dev_survey
3 ORDER BY annual_salary ASC;

```

The results are displayed in a table:

dev_id	country	experience_years	primary_language	secondary_language	database_used	framework	annual_salary
1	Pakistan	19	C++	java	sqlite3	FastApi	NULL
2	Pakistan	2	Python	JavaScript	MySQL	Django	8800
3	India	19	JavaScript	HTML	Firebase	Angular	10000
4	Pakistan	8	Java	Kotlin	MySQL	Spring	13200.0
5	India	6	PHP	JavaScript	MySQL	Laravel	15000
6	Pakistan	5	C++	Python	MySQL	NULL	15400.0
7	South Africa	7	Python	JavaScript	PostgreSQL	FastAPI	55000
8	Canada	4	JavaScript	Python	MongoDB	Vue	60000
9	France	8	Python	SQL	PostgreSQL	Django	65000
10	Netherlands	6	Go	Python	PostgreSQL	NULL	68000
11	Germany	7	C#	SQL	SQL Server	.NET	70000
12	Australia	9	JavaScript	NULL	MongoDB	React	75000
13	Japan	11	Java	Python	Oracle	Spring	82000
14	UK	6	Python	R	PostgreSQL	Flask	85000
15	USA	10	JavaScript	TypeScript	PostgreSQL	React	90000
16	Sweden	18	Python	Julia	SQLite	Django	90000
17	USA	9	C++	Python	SQLite	Qt	110000
18	Vietnam	22	NULL	PHP	NULL	NULL	117000
19	India	24	C++	java	sqlite3	FastApi	131000

Execution finished without errors. Result: 19 rows returned in 4ms At Line 1: SELECT *;

14. Show the only developer with the most experience:

Command:

```
SELECT *  
FROM dev_survey  
ORDER BY experience DESC  
LIMIT 1;
```

Output:

The screenshot shows the DB Browser for SQLite interface. The SQL tab contains the following query:

```
1 SELECT *  
2 FROM dev_survey  
3 ORDER BY experience DESC  
4 LIMIT 1;
```

The results pane displays a table with 21 rows of developer survey data. The columns are: dev_id, country, experience_years, primary_language, secondary_language, database_used, framework, and annual_salary. The data includes various developer profiles from different countries like Pakistan, India, USA, Canada, etc., with their respective experience years, primary languages (e.g., C++, Java, Python), and annual salaries.

dev_id	country	experience_years	primary_language	secondary_language	database_used	framework	annual_salary
1	Pakistan	19	C++	Java	SQLite3	FastAPI	NULL
2	Pakistan	2	Python	JavaScript	MySQL	Django	8800
3	India	2	JavaScript	HTML	Firebase	Angular	10000
4	Pakistan	3	Java	Kotlin	MySQL	Spring	13200.0
5	India	6	PHP	JavaScript	MySQL	Laravel	15000
6	Pakistan	5	C++	Python	MySQL	NULL	15400.0
7	South Africa	7	Python	JavaScript	PostgreSQL	FastAPI	55000
8	Canada	4	JavaScript	Python	MongoDB	Vue	60000
9	France	8	Python	SQL	PostgreSQL	Django	65000
10	Netherlands	6	Go	Python	PostgreSQL	NULL	68000
11	Germany	7	C#	SQL	SQL Server	.NET	70000
12	Australia	9	JavaScript	NULL	MongoDB	React	75000
13	Japan	11	Java	Python	Oracle	Spring	82000
14	UK	12	Python	R	PostgreSQL	Flask	85000
15	USA	10	JavaScript	TypeScript	PostgreSQL	React	90000
16	Sweden	10	Python	Julia	SQLite	Django	90000
17	USA	15	C++	Python	SQLite	Qt	110000
18	Vietnam	NULL	NULL	PHP	NULL	NULL	117000
19	INDIA	23	C++	Java	SQLite3	FastAPI	131000

Execution finished without errors.
Result: 21 rows returned in 4ms.
At Line 1:
SELECT *

15. Show the developer with the 4th highest salary.

Command:

```
SELECT *  
FROM developers  
ORDER BY annual_salary DESC  
LIMIT 1  
OFFSET 3;
```

Output:

The screenshot shows two windows side-by-side. The left window is a terminal window titled 'aazan-noor-khuwaja@Hp-G3: ~/Aazan_Data/4th_Semester/DB_Lab/lab3'. It displays the following SQL command and its result:

```
sqlite> SELECT * FROM developers
...> ORDER BY annual_salary DESC
Display all 157 possibilities? (y or n)
...> ORDER BY annual_salary DESC
...> LIMIT 1
...> OFFSET 3;
+-----+-----+-----+-----+-----+-----+-----+
| dev_id | country | experience_years | primary_language | secondary_language | database_used | framework | annual_salary |
+-----+-----+-----+-----+-----+-----+-----+
| 6      | UK      | 12             | Python          | R               | PostgreSQL    | Flask     | 85000       |
+-----+-----+-----+-----+-----+-----+-----+
sqlite>
```

The right window is 'DB Browser for SQLite - /home/aazan-noor-khuwaja/Aazan_Data/4th_Semester/DB_Lab/lab3/tech_survey.db'. It shows the same SQL query in the 'SQL' tab and the resulting data in the 'Data' tab. The data table has columns: dev_id, country, experience_years, primary_language, secondary_language, database_used, framework, and annual_salary. One row is visible:

dev_id	country	experience_years	primary_language	secondary_language	database_used	framework	annual_salary
6	UK	12	Python	R	PostgreSQL	Flask	85000

In the bottom status bar of the DB browser, it says: 'Execution finished without errors. Result: 1 rows returned in 17ms At line 1: SELECT *'.

16. Show developers who use both primary and secondary languages.

Command:

```
SELECT *
FROM developers
WHERE primary_language IS NOT NULL AND secondary_language IS NOT NULL;
```

Output:

```
aazan-noor-khuwaja@Hp-G3: ~/Aazan_Data/4th_Semester/DB_Lab/lab3

sqlite> SELECT * FROM developers
...> WHERE primary_language IS NOT NULL AND secondary_language IS NOT NULL;
+-----+-----+-----+-----+-----+-----+-----+-----+
| dev_id | country | experience_years | primary_language | secondary_language | database_used | framework | annual_salary |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1     | Pakistan | 2             | Python          | JavaScript       | MySQL         | Django      | 8000        | |
| 2     | India    | 5             | Java            | SQL             | Oracle        | Spring      | 18000       |
| 3     | USA      | 10            | JavaScript      | TypeScript       | PostgreSQL    | React       | 90000       |
| 4     | Germany  | 7             | C#              | SQL             | SQL Server   | .NET        | 70000       |
| 6     | UK       | 12            | Python          | R               | PostgreSQL   | Flask       | 85000       |
| 7     | Canada   | 4             | JavaScript      | Python          | MongoDB     | Vue         | 60000       |
| 8     | Pakistan | 3             | Java            | Kotlin          | MySQL        | Spring      | 12000       |
| 9     | USA      | 15            | C++             | Python          | SQLite       | Qt          | 110000      |
| 10    | India    | 6             | PHP             | JavaScript     | MySQL        | Laravel     | 15000       |
| 11    | France   | 8             | Python          | SQL             | PostgreSQL   | Django      | 65000       |
| 13    | Japan    | 11            | Java            | Python          | Oracle       | Spring      | 82000       |
| 14    | Brazil   | 4             | Python          | JavaScript     | MySQL        | Flask       | 22000       |
| 15    | Netherlands | 6             | Go              | Python          | PostgreSQL   | PostgreSQL  | Rails       | 68000       |
| 16    | USA      | 3             | Ruby             | JavaScript     | PostgreSQL   | MySQL       | 72000       |
| 17    | Pakistan | 5             | C++             | Python          | MySQL        | SQLite      | 14000       |
| 18    | Sweden   | 10            | Python          | Julia           | Julia        | Django      | 90000       |
| 19    | India    | 2             | JavaScript      | HTML            | Firebase     | Angular     | 10000       |
| 20    | South Africa | 7             | Python          | JavaScript     | PostgreSQL   | FastAPI    | 55000       |
+-----+-----+-----+-----+-----+-----+-----+-----+
sqlite>
```

DB Browser for SQLite - /home/aazan-noor-khuwaja/Aazan_Data/4th_Semester/DB_Lab/lab3/tech_survey.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1

```
1 SELECT *
2 FROM developers
3 WHERE primary_language IS NOT NULL AND secondary_language IS NOT NULL;
```

dev_id	country	experience_years	primary_language	secondary_language	database_used	framework	annual_salary
1	Pakistan	2	Python	JavaScript	MySQL	Django	8000
2	India	5	Java	SQL	Oracle	Spring	18000
3	USA	10	JavaScript	TypeScript	PostgreSQL	React	90000
4	Germany	7	C#	SQL	SQL Server	.NET	70000
6	UK	12	Python	R	PostgreSQL	Flask	85000
7	Canada	4	JavaScript	Python	MongoDB	Vue	60000
8	Pakistan	3	Java	Kotlin	MySQL	Spring	12000
9	USA	15	C++	Python	SQLite	Qt	110000
10	India	6	PHP	JavaScript	MySQL	Laravel	15000
11	France	8	Python	SQL	PostgreSQL	Django	65000
13	Japan	11	Java	Python	Oracle	Spring	82000
14	Brazil	4	Python	JavaScript	MySQL	Flask	22000
15	Netherlands	6	Go	Python	PostgreSQL	PostgreSQL	68000
16	USA	3	Ruby	JavaScript	PostgreSQL	Rails	72000
17	Pakistan	5	C++	Python	MySQL	Rails	14000
18	Sweden	10	Python	Julia	SQLite	Django	90000
19	India	2	JavaScript	HTML	Firebase	Angular	10000
20	South Africa	7	Python	JavaScript	PostgreSQL	FastAPI	55000

Execution finished without errors.
Result: 18 rows returned in 21ms
At line 1:
SELECT *
FROM developers
WHERE primary_language IS NOT NULL AND secondary_language IS NOT NULL;

17. Show developers where salary information is not provided.

Command:

SELECT *

FROM developers

WHERE annual_salary IS NULL;

Output:

```
aazan-noor-khuwaja@Hp-G3: ~/Aazan_Data/4th_Semester/DB_Lab/lab3
sqlite> SELECT * FROM developers
...> WHERE annual_salary IS NULL;
+-----+-----+-----+-----+-----+-----+-----+
| dev_id | country | experience_years | primary_language | secondary_language | database_used | framework | annual_salary |
+-----+-----+-----+-----+-----+-----+-----+
| 5      | Pakistan | 1              | C                | NULL             | NULL          | NULL       | NULL        |
+-----+-----+-----+-----+-----+-----+-----+
sqlite>
```

DB Browser for SQLite - /home/aazan:/

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project

Database Structure Browse Data Edit Pragmas Execute SQL

Table: **developers** Filter in any column

	dev_id	country	experience_years	primary_language	secondary_language	database_used	framework	annual_salary
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	Pakistan		2 Python	JavaScript	MySQL	Django	8000
2	2	India		5 Java	SQL	Oracle	Spring	18000
3	3	USA		10 JavaScript	TypeScript	PostgreSQL	React	90000
4	4	Germany		7 C#	SQL	SQL Server	.NET	70000
5	5	Pakistan		1 C	NULL	NULL	NULL	NULL
6	6	UK		12 Python	R	PostgreSQL	Flask	85000
7	7	Canada		4 JavaScript	Python	MongoDB	Vue	60000
8	8	Pakistan		3 Java	Kotlin	MySQL	Spring	12000
9	9	USA		15 C++	Python	SQLite	Qt	110000
10	10	India		6 PHP	JavaScript	MySQL	Laravel	15000
11	11	France		8 Python	SQL	PostgreSQL	Django	65000
12	12	Australia		9 JavaScript	NULL	MongoDB	React	75000
13	13	Japan		11 Java	Python	Oracle	Spring	82000
14	14	Brazil		4 Python	JavaScript	MySQL	Flask	22000
15	15	Netherlands		6 Go	Python	PostgreSQL	NULL	68000
16	16	USA		3 Ruby	JavaScript	PostgreSQL	Rails	72000
17	17	Pakistan		5 C++	Python	MySQL	NULL	14000
18	18	Sweden		10 Python	Julia	SQLite	Django	90000
19	19	India		2 JavaScript	HTML	Firebase	Angular	10000
20	20	South Africa		7 Python	JavaScript	PostgreSQL	FastAPI	55000

Figure 1 There was no null in annual salary so for that i added one with null to show some output .

The screenshot shows the DB Browser for SQLite interface. The title bar reads "DB Browser for SQLite - /home/aaazan-noor-khwaja/Aazan_Data/4th_Semester/DB_Lab/lab3/tech_survey.db". The menu bar includes File, Edit, View, Tools, Help, New Database, Open Database, Write Changes, Revert Changes, Open Project, Save Project, Attach Database, and Close Database. Below the menu is a toolbar with icons for New Database, Open Database, Write Changes, Revert Changes, Open Project, Save Project, Attach Database, and Close Database. A tab bar at the top right shows Database Structure, Browse Data, Edit Pragmas, and Execute SQL, with Execute SQL selected. The main area has a toolbar with icons for Open, Save, Print, Copy, Paste, Find, Replace, and others. A SQL editor window titled "SQL 1" contains the following code:

```
1 SELECT *
2 FROM developers
3 WHERE annual_salary IS NULL;
```

Below the SQL editor is a results grid with the following data:

dev_id	country	experience_years	primary_language	secondary_language	database_used	framework	annual_salary
1	Pakistan	1	C	NULL	NULL	NULL	NULL

At the bottom of the results grid, the message "Execution finished without errors. Result: 1 rows returned in 20ms At line 1: SELECT * FROM developers WHERE annual_salary IS NULL;" is displayed.

18. For each country (ordered alphabetically), rank developers from highest to lowest salary.

Command:

```
SELECT *
FROM developers
ORDER BY country ASC,
annual_salary DESC;
```

Output:

```

aazan-noor-khuwaja@Hp-G3: ~/Aazan_Data/4th_Semester/DB_Lab/lab3
sqlite> SELECT * FROM developers
...> ORDER BY country ASC, annual_salary DESC;
-----+-----+-----+-----+-----+-----+-----+
| dev_id | country | experience_years | primary_language | secondary_language | database_used | framework | annual_salary |
-----+-----+-----+-----+-----+-----+-----+
| 12 | Australia | 9 | JavaScript | NULL | MongoDB | React | 75000 |
| 14 | Brazil | 4 | Python | JavaScript | MySQL | Flask | 22000 |
| 7 | Canada | 4 | JavaScript | Python | MongoDB | Vue | 60000 |
| 11 | France | 8 | Python | SQL | PostgreSQL | Django | 65000 |
| 4 | Germany | 7 | C# | SQL | SQL Server | .NET | 70000 |
| 2 | India | 5 | Java | SQL | Oracle | Spring | 18000 |
| 10 | India | 6 | PHP | JavaScript | MySQL | Laravel | 15000 |
| 19 | India | 2 | JavaScript | HTML | Firebase | Angular | 10000 |
| 13 | Japan | 11 | Java | Python | Oracle | Spring | 82000 |
| 15 | Netherlands | 6 | Go | Python | PostgreSQL | MySQL | 68000 |
| 17 | Pakistan | 5 | C++ | Python | MySQL | MySQL | 14000 |
| 8 | Pakistan | 3 | Java | Kotlin | MySQL | Spring | 12000 |
| 1 | Pakistan | 2 | Python | JavaScript | MySQL | Django | 8000 |
| 5 | Pakistan | 1 | C | NULL | NULL | NULL | NULL |
| 20 | South Africa | 7 | Python | JavaScript | PostgreSQL | FastAPI | 55000 |
| 18 | Sweden | 10 | Python | Julia | SQLite | Django | 90000 |
| 6 | UK | 12 | Python | R | PostgreSQL | Flask | 85000 |
| 9 | USA | 15 | C++ | Python | SQLite | Qt | 110000 |
| 3 | USA | 10 | JavaScript | TypeScript | PostgreSQL | React | 90000 |
| 16 | USA | 3 | Ruby | JavaScript | PostgreSQL | Rails | 72000 |
-----+-----+-----+-----+-----+-----+-----+

```

DB Browser for SQLite - /home/aazan-noor-khuwaja/Aazan_Data/4th_Semester/DB_Lab/lab3/tech_survey.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1

```

1 SELECT *
2 FROM developers
3 ORDER BY country ASC,
4 annual_salary DESC;

```

dev_id	country	experience_years	primary_language	secondary_language	database_used	framework	annual_salary
1	Australia	9	JavaScript	NULL	MongoDB	React	75000
2	Brazil	4	Python	JavaScript	MySQL	Flask	22000
3	Canada	4	JavaScript	Python	MongoDB	Vue	60000
4	France	8	Python	SQL	PostgreSQL	Django	65000
5	Germany	7	C#	SQL	SQL Server	.NET	70000
6	India	5	Java	SQL	Oracle	Spring	18000
10	India	6	PHP	JavaScript	MySQL	Laravel	15000
19	India	2	JavaScript	HTML	Firebase	Angular	10000
13	Japan	11	Java	Python	Oracle	Spring	82000
15	Netherlands	6	Go	Python	PostgreSQL	MySQL	68000
17	Pakistan	5	C++	Python	MySQL	MySQL	14000
8	Pakistan	3	Java	Kotlin	MySQL	Spring	12000
1	Pakistan	2	Python	JavaScript	MySQL	Django	8000
5	Pakistan	1	C	NULL	NULL	NULL	NULL
20	South Africa	7	Python	JavaScript	PostgreSQL	FastAPI	55000
18	Sweden	10	Python	Julia	SQLite	Django	90000
6	UK	12	Python	R	PostgreSQL	Flask	85000
9	USA	15	C++	Python	SQLite	Qt	110000
3	USA	10	JavaScript	TypeScript	PostgreSQL	React	90000
16	USA	3	Ruby	JavaScript	PostgreSQL	Rails	72000

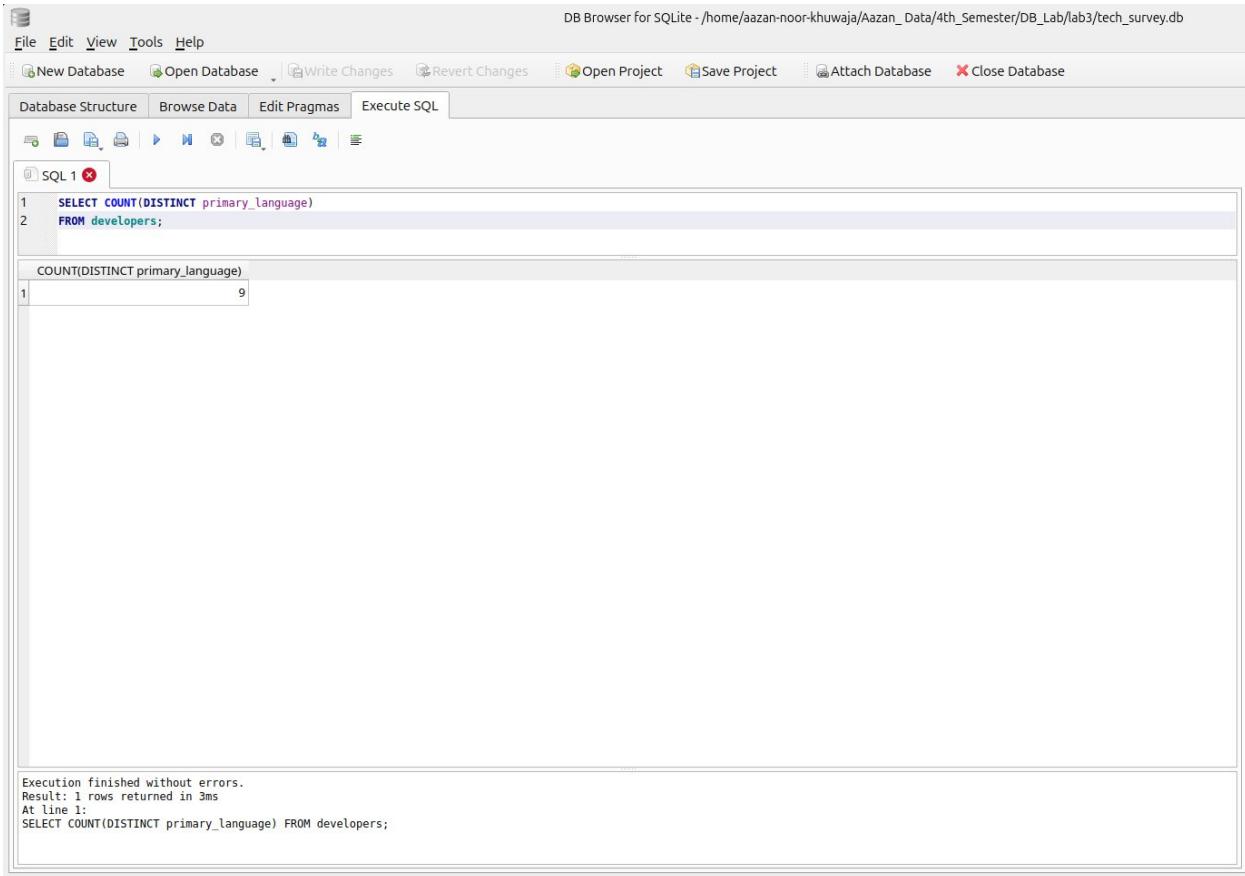
Execution finished without errors.
Result: 20 rows returned in 38ms
At line 1:
SELECT *
FROM developers

19. Display the total number of unique primary programming languages used by all developers.

Command:

```
SELECT COUNT(DISTINCT primary_language)
FROM developers;
```

Output:



The screenshot shows the DB Browser for SQLite interface. The title bar indicates the database is 'tech_survey.db'. The menu bar includes File, Edit, View, Tools, and Help. The toolbar has icons for New Database, Open Database, Write Changes, Revert Changes, Open Project, Save Project, Attach Database, and Close Database. Below the toolbar are tabs for Database Structure, Browse Data, Edit Pragmas, and Execute SQL. The Execute SQL tab is active, showing a single query in the SQL editor: 'SELECT COUNT(DISTINCT primary_language) FROM developers;'. The results pane displays a table with one row: COUNT(DISTINCT primary_language) = 9. At the bottom of the results pane, the message 'Execution finished without errors. Result: 1 rows returned in 3ms' is shown, along with the executed SQL command.

20. Display developers ranked 6th to 10th by salary, where rank 1 corresponds to the highest salary overall.

Command:

```
SELECT *
FROM developers
ORDER BY annual_salary DESC
LIMIT 5
OFFSET 5;
```

Output:

```

sqlite> SELECT * FROM developers
...> ORDER BY annual_salary DESC
...> LIMIT 5
...> OFFSET 5;
+-----+-----+-----+-----+-----+-----+-----+-----+
| dev_id | country | experience_years | primary_language | secondary_language | database_used | framework | annual_salary |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 12 | Australia | 9 | JavaScript | NULL | MongoDB | React | 75000 |
| 16 | USA | 3 | Ruby | JavaScript | PostgreSQL | Rails | 72000 |
| 4 | Germany | 7 | C# | SQL | SQL Server | .NET | 70000 |
| 15 | Netherlands | 6 | Go | Python | PostgreSQL | NULL | 68000 |
| 11 | France | 8 | Python | SQL | PostgreSQL | Django | 65000 |
+-----+-----+-----+-----+-----+-----+-----+-----+
sqlite> 

```

The screenshot shows the DB Browser for SQLite interface. The title bar reads "DB Browser for SQLite - /home/aazan-noor-khuwaja/Aazan_Data/4th_Semester/DB_Lab/lab3/tech_survey.db". The main window has a toolbar with File, Edit, View, Tools, Help, New Database, Open Database, Write Changes, Revert Changes, Open Project, Save Project, Attach Database, and Close Database. Below the toolbar are tabs for Database Structure, Browse Data, Edit Pragmas, and Execute SQL. The Execute SQL tab is active, containing the following SQL code:

```

1 SELECT *
2 FROM developers
3 ORDER BY annual_salary DESC
4 LIMIT 5;
5 OFFSET 5;
6

```

Below the SQL editor is a table view showing the results of the query. The columns are: dev_id, country, experience_years, primary_language, secondary_language, database_used, framework, and annual_salary. The data is identical to the one shown in the terminal output above.

Execution finished without errors.
Result: 5 rows returned in 29ms
At line 1:
SELECT *

21. Display the average annual salary of developers whose primary programming language is Java and whose database is Oracle.

Command:

SELECT AVG(annual_salary)

FROM developers

WHERE primary_language='Java' AND database_used= 'Oracle';

Output:

```

39.2% 0.89 21.2% 214.8 B/s 4.6 KB/s 35.0°C
Wed Feb 4 12:06 PM

aazan-noor-khuwaja@Hp-G3: ~/Aazan_Data/4th_Semester/DB_Lab/lab3

sqlite> SELECT AVG(annual_salary) FROM developers
...> WHERE primary_language = 'Java' AND secondary_language = 'Oracle';
+-----+
| AVG(annual_salary) |
+-----+
|           |
+-----+
sqlite> SELECT AVG(annual_salary) FROM developers
...> WHERE primary_language='Java' AND database_used= 'Oracle';
+-----+
| AVG(annual_salary) |
+-----+
| 50000.0          |
+-----+
sqlite>

```

DB Browser for SQLite - /home/aazan-noor-khuwaja/aazan_Data/4th_Semester/DB_Lab/lab3/tech_survey.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1

```

1 SELECT AVG(annual_salary)
2 FROM developers
3 WHERE primary_language='Java' AND database_used= 'Oracle';

```

AVG(annual_salary)
50000.0

Execution finished without errors.
Result: 1 rows returned in 1ms
At line 1:
SELECT AVG(annual_salary)

22. Display a new calculated column named bonus, where the bonus is computed for each developer as : bonus = annual_salary + (1000 × experience_years).

Command:

```
SELECT *,annual_salary+(1000*experience_years) AS bonus
```

```
FROM developers;
```

Output:

```
aazan-noor-khuwaja@Hp-G3: ~/Aazan_Data/4th_Semester/DB_Lab/lab3

SELECT * , annual_salary(1000*experience_years) AS bonus FROM developers ;
      ^--- error here
sqlite> SELECT * , annual_salary(1000*experience_years) AS bonus
...> FROM developers
...> ;
Parse error: no such function: annual_salary
SELECT * , annual_salary(1000*experience_years) AS bonus FROM developers ;
      ^--- error here
sqlite> SELECT *
...> annual_salary+(1000 * experience_years) AS bonus
...> FROM developers ;
Parse error: near "annual_salary": syntax error
SELECT * annual_salary+(1000 * experience_years) AS bonus FROM developers ;
      ^--- error here
sqlite> SELECT *
...> , annual_salary+(1000 * experience_years) AS bonus
...> FROM developers ;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| dev_id | country | experience_years | primary_language | secondary_language | database_used | framework | annual_salary | bonus |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1     | Pakistan | 2           | Python          | JavaScript        | MySQL          | Django       | 8000         | 10000    | |
| 2     | India    | 5           | Java            | SQL              | Oracle         | Spring       | 18000        | 23000    |
| 3     | USA      | 10          | JavaScript      | TypeScript        | PostgreSQL     | React        | 90000        | 100000   |
| 4     | Germany  | 7           | C#              | SQL              | SQL Server    | .NET         | 70000        | 77000    |
| 5     | Pakistan | 1           | C               |                   |               |             |             |             |
| 6     | UK       | 12          | Python          | R                | PostgreSQL    | Flask        | 85000        | 97000    |
| 7     | Canada   | 4           | JavaScript      | Python           | MongoDB       | Vue          | 60000        | 64000    |
| 8     | Pakistan | 3           | Java            | Kotlin           | MySQL          | Spring       | 12000        | 15000    |
| 9     | USA      | 15          | C++             | Python           | SQLite         | Qt           | 110000       | 125000   |
| 10    | India    | 6           | PHP             | JavaScript       | MySQL          | Laravel      | 15000        | 21000    |
| 11    | France   | 8           | Python          | SQL              | PostgreSQL    | Django       | 65000        | 73000    |
| 12    | Australia | 9           | JavaScript      |                   | MySQL          | MongoDB     | React        | 75000        | 84000    |
| 13    | Japan    | 11          | Java            | Python           | Oracle         | Spring       | 82000        | 93000    |
| 14    | Brazil   | 4           | Python          | Python           | MySQL          | Flask        | 22000        | 26000    |
| 15    | Netherlands | 6           | Go              | Python           | PostgreSQL    | PostgreSQL   | Rails        | 72000        | 75000    |
| 16    | USA      | 3           | Ruby             | JavaScript       | PostgreSQL    | MySQL        | Django       | 14000        | 19000    |
| 17    | Pakistan | 5           | C++             | Python           | MySQL          | Julia        | SQLite       | 90000        | 100000   |
| 18    | Sweden   | 10          | Python          | Julia            | Firebase      | Angular      | 10000        | 12000    |
| 19    | India    | 2           | JavaScript      | HTML             | Firebase      | PostgreSQL   | FastAPI     | 55000        | 62000    |
| 20    | South Africa | 7           | Python          | JavaScript       | PostgreSQL   | PostgreSQL  |             |             |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
sqlite>
```

DB Browser for SQLite - /home/aazan-noor-khuwaja/Aazan_Data/4th_Semester/DB_Lab/lab3/tech_survey.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1

```
1 SELECT *,annual_salary+(1000*experience_years) AS bonus
2 FROM developers;
3
```

	dev_id	country	experience_years	primary_language	secondary_language	database_used	framework	annual_salary	bonus
1	1	Pakistan	2	Python	JavaScript	MySQL	Django	8000	10000
2	2	India	5	Java	SQL	Oracle	Spring	18000	23000
3	3	USA	10	JavaScript	TypeScript	PostgreSQL	React	90000	100000
4	4	Germany	7	C#	SQL	SQL Server	.NET	70000	77000
5	5	Pakistan	1	C	NULL	NULL	NULL	NULL	NULL
6	6	UK	12	Python	R	PostgreSQL	Flask	85000	97000
7	7	Canada	4	JavaScript	Python	MongoDB	Vue	60000	64000
8	8	Pakistan	3	Java	Kotlin	MySQL	Spring	12000	15000
9	9	USA	15	C++	Python	SQLite	Qt	110000	125000
10	10	India	6	PHP	JavaScript	MySQL	Laravel	15000	21000
11	11	France	8	Python	SQL	PostgreSQL	Django	65000	73000
12	12	Australia	9	JavaScript	NULL	MongoDB	React	75000	84000
13	13	Japan	11	Java	Python	Oracle	Spring	82000	93000
14	14	Brazil	4	Python	JavaScript	MySQL	Flask	22000	26000
15	15	Netherlands	6	Go	Python	PostgreSQL	NULL	68000	74000
16	16	USA	3	Ruby	JavaScript	PostgreSQL	Rails	72000	75000
17	17	Pakistan	5	C++	Python	MySQL	NULL	14000	19000
18	18	Sweden	10	Python	Julia	SQLite	Django	90000	100000
19	19	India	2	JavaScript	HTML	Firebase	Angular	10000	12000

Execution finished without errors.
Result: 20 rows returned in 5ms
At line 1:
SELECT *,annual_salary+(1000*experience_years) AS bonus
FROM developers;