

QUERYING DATA FROM SINGLE TABLE

Create Table:

```
1 INSERT INTO studentsinfo (StudentId, StudentName, City, PostCode, Income)
2 VALUES (1, 'Alice Johnson', 'New York', '10001', 45000),
3 (2, 'Bob Smith', 'Los Angeles', '90001', 52000),
4 (3, 'Carol Williams', 'Chicago', '60601', 47000),
5 (4, 'David Brown', 'Houston', '77001', 48000),
6 (5, 'Emma Davis', 'Phoenix', '85001', 50000),
7 (6, 'Frank Miller', 'Philadelphia', '19101', 53000),
8 (7, 'Grace Wilson', 'San Antonio', '78201', 46000),
9 (8, 'Henry Moore', 'San Diego', '92101', 51000),
10 (9, 'Isabella Taylor', 'Dallas', '75201', 49500),
11 (10, 'Jack Anderson', 'San Jose', '95101', 52500);
```

StudentId	StudentName	City	PostCode	Income
1	Alice Johnson	New York	10001	45000
1	Alice Johnson	New York	10001	45000
2	Bob Smith	Los Angeles	90001	52000
3	Carol Williams	Chicago	60601	47000
4	David Brown	Houston	77001	48000
5	Emma Davis	Phoenix	85001	50000
6	Frank Miller	Philadelphia	19101	53000
7	Grace Wilson	San Antonio	78201	46000
8	Henry Moore	San Diego	92101	51000
9	Isabella Taylor	Dallas	75201	49500
10	Jack Anderson	San Jose	95101	52500

SELECT statement is used to select data from a database:

```
SELECT * from studentinfo;
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Run](#)]

☐ Show all | Number of rows: 25 ▼ Filter rows:

Extra options

StudentId	StudentName	City	PostCode	Income
1	Alice Johnson	New York	10001	45000
2	Bob Smith	Los Angeles	90001	52000
3	Carol Williams	Chicago	60601	47000
4	David Brown	Houston	77001	48000
5	Emma Davis	Phoenix	85001	50000
6	Frank Miller	Philadelphia	19101	53000
7	Grace Wilson	San Antonio	78201	46000
8	Henry Moore	San Diego	92101	51000
9	Isabella Taylor	Dallas	75201	49500
10	Jack Anderson	San Jose	95101	52500

SELECT DISTINCT statement is used to return only distinct (different) values:

```
SELECT DISTINCT City FROM studentinfo;
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Run](#)]

☐ Show all | Number of rows: 25 ▼

Extra options

City

New York

Los Angeles

Chicago

Houston

Phoenix

Philadelphia

San Antonio

San Diego

Dallas

San Jose

Your SQL query has been executed successfully.

```
SELECT COUNT(DISTINCT City) FROM studentinfo;
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create F](#)]

Extra options

COUNT(DISTINCT City)

10

WHERE clause is used to filter records:

✓ Showing rows 0 - 0 (1 total, Query took 0.0010 seconds.)

```
SELECT * FROM studentinfo WHERE City = 'New York';
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Ref](#)

☐ Show all | Number of rows: 25 ▼ Filter rows:

Extra options

StudentId	StudentName	City	PostCode	Income
1	Alice Johnson	New York	10001	45000

✓ Showing rows 0 - 0 (1 total, Query took 0.0143 seconds.)

```
SELECT * FROM studentinfo WHERE StudentId = 10;
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Ref](#)

☐ Show all | Number of rows: 25 ▼ Filter rows:

Extra options

StudentId	StudentName	City	PostCode	Income
10	Jack Anderson	San Jose	95101	52500

WHERE clause can be combined with AND, OR, and NOT operators:

✓ Showing rows 0 - 0 (1 total, Query took 0.0004 seconds.)

```
SELECT * FROM studentinfo WHERE City = 'Chicago' AND Age = 22;
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

☐ Show all | Number of rows: 25 ▼ Filter rows:

Extra options

StudentId	StudentName	City	PostCode	Income	Age	CGPA
3	Carol Williams	Chicago	60601	47000	22	3.90

✓ Showing rows 0 - 2 (3 total, Query took 0.0003 seconds.)

```
SELECT * FROM studentinfo WHERE City = 'Dallas' OR Age = 20;
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

☐ Show all | Number of rows: 25 ▼ Filter rows:

Extra options

StudentId	StudentName	City	PostCode	Income	Age	CGPA
1	Alice Johnson	New York	10001	45000	20	3.80
7	Grace Wilson	San Antonio	78201	46000	20	3.80
9	Isabella Taylor	Dallas	75201	49500	22	3.90

✓ Showing rows 0 - 8 (9 total, Query took 0.0123 seconds.)

```
SELECT * FROM studentinfo WHERE NOT City = 'Philadelphia';
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

☐ Show all | Number of rows: 25 ▼ Filter rows:

Extra options

StudentId	StudentName	City	PostCode	Income	Age	CGPA
1	Alice Johnson	New York	10001	45000	20	3.80
2	Bob Smith	Los Angeles	90001	52000	21	3.50
3	Carol Williams	Chicago	60601	47000	22	3.90
4	David Brown	Houston	77001	48000	23	3.60
5	Emma Davis	Phoenix	85001	50000	24	3.70
7	Grace Wilson	San Antonio	78201	46000	20	3.80
8	Henry Moore	San Diego	92101	51000	21	3.50
9	Isabella Taylor	Dallas	75201	49500	22	3.90
10	Jack Anderson	San Jose	95101	52500	23	3.60

✓ Showing rows 0 - 0 (1 total, Query took 0.0003 seconds.)

```
SELECT * FROM studentinfo WHERE City = 'New York' AND (CGPA >= 3.5 OR Age = 22);
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

☐ Show all | Number of rows: 25 ▼ Filter rows:

Extra options

StudentId	StudentName	City	PostCode	Income	Age	CGPA
1	Alice Johnson	New York	10001	45000	20	3.80

ORDER BY keyword is used to sort the result-set in ascending or descending order:

```
SELECT * FROM studentinfo ORDER BY City;
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[F](#)

☐ Show all | Number of rows: Filter rows:

Extra options

StudentId	StudentName	City ▲ 1	PostCode	Income	Age	CGPA
3	Carol Williams	Chicago	60601	47000	22	3.90
9	Isabella Taylor	Dallas	75201	49500	22	3.90
4	David Brown	Houston	77001	48000	23	3.60
2	Bob Smith	Los Angeles	90001	52000	21	3.50
1	Alice Johnson	New York	10001	45000	20	3.80
6	Frank Miller	Philadelphia	19101	53000	25	3.40
5	Emma Davis	Phoenix	85001	50000	24	3.70
7	Grace Wilson	San Antonio	78201	46000	20	3.80
8	Henry Moore	San Diego	92101	51000	21	3.50
10	Jack Anderson	San Jose	95101	52500	23	3.60

```
SELECT * FROM studentinfo ORDER BY City DESC;
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[F](#)

☐ Show all | Number of rows: Filter rows:

Extra options

StudentId	StudentName	City ▼ 1	PostCode	Income	Age	CGPA
10	Jack Anderson	San Jose	95101	52500	23	3.60
8	Henry Moore	San Diego	92101	51000	21	3.50
7	Grace Wilson	San Antonio	78201	46000	20	3.80
5	Emma Davis	Phoenix	85001	50000	24	3.70
6	Frank Miller	Philadelphia	19101	53000	25	3.40
1	Alice Johnson	New York	10001	45000	20	3.80
2	Bob Smith	Los Angeles	90001	52000	21	3.50
4	David Brown	Houston	77001	48000	23	3.60
9	Isabella Taylor	Dallas	75201	49500	22	3.90
3	Carol Williams	Chicago	60601	47000	22	3.90

UPDATE statement is used to modify the existing records in a table:

```
UPDATE studentinfo SET StudentName = 'Faysal', City = 'Dhaka', PostCode = '1361' WHERE StudentId = 1;
```

StudentId	StudentName	City	PostCode	Income	Age	CGPA
1	Faysal	Dhaka	1361	45000	20	3.80
2	Bob Smith	Los Angeles	90001	52000	21	3.50
3	Carol Williams	Chicago	60601	47000	22	3.90
4	David Brown	Houston	77001	48000	23	3.60
5	Emma Davis	Phoenix	85001	50000	24	3.70
6	Frank Miller	Philadelphia	19101	53000	25	3.40
7	Grace Wilson	San Antonio	78201	46000	20	3.80
8	Henry Moore	San Diego	92101	51000	21	3.50
9	Isabella Taylor	Dallas	75201	49500	22	3.90
10	Jack Anderson	San Jose	95101	52500	23	3.60

DELETE statement is used to delete existing records in a table:

```
DELETE FROM studentinfo WHERE StudentName = 'Frank Miller';
```

StudentId	StudentName	City	PostCode	Income	Age	CGPA
1	Faysal	Dhaka	1361	45000	20	3.80
2	Bob Smith	Los Angeles	90001	52000	21	3.50
3	Carol Williams	Chicago	60601	47000	22	3.90
4	David Brown	Houston	77001	48000	23	3.60
5	Emma Davis	Phoenix	85001	50000	24	3.70
7	Grace Wilson	San Antonio	78201	46000	20	3.80
8	Henry Moore	San Diego	92101	51000	21	3.50
9	Isabella Taylor	Dallas	75201	49500	22	3.90
10	Jack Anderson	San Jose	95101	52500	23	3.60

BETWEEN operator selects values within a given range:

```
SELECT * FROM studentinfo WHERE Income BETWEEN 45000 AND 50000;
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)]

☐ Show all | Number of rows: 25 ▼ Filter rows:

Extra options

StudentId	StudentName	City	PostCode	Income	Age	CGPA
1	Faysal	Dhaka	1361	45000	20	3.80
3	Carol Williams	Chicago	60601	47000	22	3.90
4	David Brown	Houston	77001	48000	23	3.60
5	Emma Davis	Phoenix	85001	50000	24	3.70
7	Grace Wilson	San Antonio	78201	46000	20	3.80
9	Isabella Taylor	Dallas	75201	49500	22	3.90

MIN() and MAX() function returns the smallest & largest value:

```
SELECT MAX(Income) FROM studentinfo;
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)]

☐ Show all | Number of rows: 25 ▼

Extra options

MAX(Income)

52500

```
SELECT MIN(Income) FROM studentinfo;
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)]

☐ Show all | Number of rows: 25 ▼

Extra options

MIN(Income)

45000

Inner Join:

```
SELECT students.StudentId, students.City, university.StudentName,  
university.UniversityName FROM students INNER JOIN university ON  
students.StudentName = university.StudentName;
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code]

☐ Show all | Number of rows: 25 ▼ Filter rows: Search this table

Extra options

StudentId	City	StudentName	UniversityName
1	Dhaka	Faysal	BUET
2	Los Angeles	Bob Smith	BRAC
3	Chicago	Carol Williams	NSU
4	Houston	David Brown	AIUB
5	Phoenix	Emma Davis	AUST
6	Philadelphia	Frank Miller	UPenn
7	San Antonio	Grace Wilson	UTSA
8	San Diego	Henry Moore	UCSD
9	Dallas	Isabella Taylor	SMU
10	San Jose	Jack Anderson	SJSU

Left Join:

```
SELECT students.StudentId, students.City, university.StudentName FROM  
students LEFT JOIN university ON students.StudentName =  
university.StudentName;
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code]

☐ Show all | Number of rows: 25 ▼ Filter rows: Search this table

Extra options

StudentId	City	StudentName
1	Dhaka	Faysal
2	Los Angeles	Bob Smith
3	Chicago	Carol Williams
4	Houston	David Brown
5	Phoenix	Emma Davis
6	Philadelphia	Frank Miller
7	San Antonio	Grace Wilson
8	San Diego	Henry Moore
9	Dallas	Isabella Taylor
10	San Jose	Jack Anderson

Right Join:

```
SELECT students.StudentId, students.City, university.StudentName FROM
students RIGHT JOIN university ON students.StudentName =
university.StudentName;
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code]

☐ Show all | Number of rows: 25 ▼ Filter rows: Search this table

Extra options

StudentId	City	StudentName
1	Dhaka	Faysal
2	Los Angeles	Bob Smith
3	Chicago	Carol Williams
4	Houston	David Brown
5	Phoenix	Emma Davis
6	Philadelphia	Frank Miller
7	San Antonio	Grace Wilson
8	San Diego	Henry Moore
9	Dallas	Isabella Taylor
10	San Jose	Jack Anderson

Full outer Join:

```
SELECT students.StudentName, students.City, university.StudentName,
university.UniversityName FROM students LEFT JOIN university ON
students.StudentName = university.StudentName UNION SELECT
students.StudentName, students.City, university.StudentName,
university.UniversityName FROM students RIGHT JOIN university ON
students.StudentName = university.StudentName;
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code]

☐ Show all | Number of rows: 25 ▼ Filter rows: Search this table

Extra options

StudentName	City	StudentName	UniversityName
Faysal	Dhaka	Faysal	BUET
Bob Smith	Los Angeles	Bob Smith	BRAC
Carol Williams	Chicago	Carol Williams	NSU
David Brown	Houston	David Brown	AIUB
Emma Davis	Phoenix	Emma Davis	AUST
Frank Miller	Philadelphia	Frank Miller	UPenn
Grace Wilson	San Antonio	Grace Wilson	UTSA
Henry Moore	San Diego	Henry Moore	UCSD
Isabella Taylor	Dallas	Isabella Taylor	SMU
Jack Anderson	San Jose	Jack Anderson	SJSU