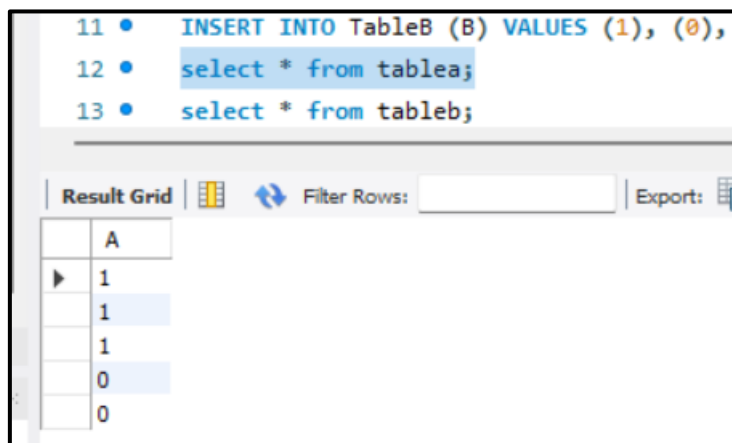


SQL-JOINS

```
use sql_joins;
show tables;
CREATE TABLE TableA (
  A INT
);
INSERT INTO TableA (A) VALUES (1), (1), (1), (0), (0);

CREATE TABLE TableB (
  B INT
);
INSERT INTO TableB (B) VALUES (1), (0), (1);
```

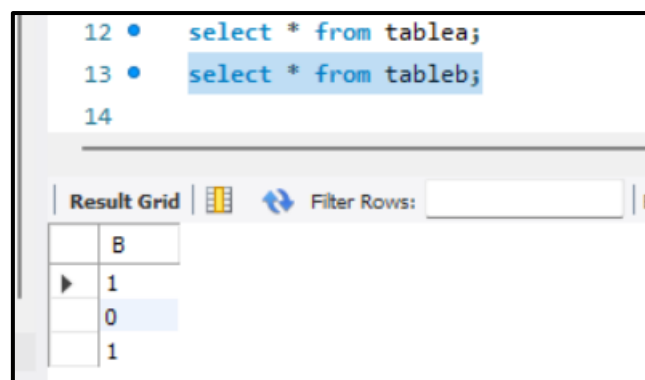
```
select * from tablea;
```



The screenshot shows a SQL query editor with three lines of code: line 11: `INSERT INTO TableB (B) VALUES (1), (0),`; line 12: `select * from tablea;` (highlighted); line 13: `select * from tableb;`. Below the editor is a 'Result Grid' showing the results of the query on line 12. The grid has a header row with 'A' and five data rows with values 1, 1, 1, 0, and 0. The interface includes a 'Filter Rows' input and an 'Export' button.

A
1
1
1
0
0

```
select * from tableb;
```



The screenshot shows a SQL query editor with three lines of code: line 12: `select * from tablea;`; line 13: `select * from tableb;` (highlighted); line 14: (empty). Below the editor is a 'Result Grid' showing the results of the query on line 13. The grid has a header row with 'B' and three data rows with values 1, 0, and 1. The interface includes a 'Filter Rows' input and an 'Export' button.

B
1
0
1

Select * from tablea left join tableb on tablea.A = tableb.B;

```

11 • INSERT INTO TableB (B) VALUES (1), (0), (1);
12 • select * from tablea;
13 • select * from tableb;
14
15 • Select * from tablea left join tableb on tablea.A = tableb.B;
16 • Select * from tablea right join tableb on tablea.A = tableb.B;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	A	B
▶	1	1
	1	1
	1	1
	1	1
	1	1
	1	1
	0	0
	0	0

Select * from tablea right join tableb on tablea.A = tableb.B;

```

14
15 • Select * from tablea left join tableb on tablea.A = tableb.B;
16 • Select * from tablea right join tableb on tablea.A = tableb.B;
17 • Select * from tablea inner join tableb on tablea.A = tableb.B;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	A	B
▶	1	1
	1	1
	1	1
	0	0
	0	0
	1	1
	1	1
	1	1

Select * from tablea inner join tableb on tablea.A = tableb.B;

```
15 • Select * from tablea left join tableb on tablea.A = tableb.B;
16 • Select * from tablea right join tableb on tablea.A = tableb.B;
17 • Select * from tablea inner join tableb on tablea.A = tableb.B;
18 • Select * from tablea full join tableb on tablea.A = tableb.B;
```

Result Grid | | Filter Rows: | Export: | Wrap Cell Content:

	A	B
▶	1	1
	1	1
	1	1
	1	1
	1	1
	1	1
	0	0
	0	0

Select * from tablea union Select * from tableb;

```
18
19 • select * from tablea union select * from tableb;
20
```

Result Grid | | Filter Rows: | Export: | Wrap Cell Cor

	A
▶	1
	0

Notes:

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
ORDER BY column_name(s);
```

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
HAVING condition
ORDER BY column_name(s);
```

```
SELECT column_name(s)
FROM table_name
WHERE EXISTS
(SELECT column_name FROM table_name WHERE condition);
```

```
SELECT COUNT(column_name)
FROM table_name
WHERE condition;
```

```
SELECT SUM(column_name)
FROM table_name
WHERE condition;
```

```
SELECT AVG(column_name)
FROM table_name
WHERE condition;
```

```
SELECT DISTINCT column1, column2, ...
FROM table_name;
```

- **MIN()** - returns the smallest value within the selected column
- **MAX()** - returns the largest value within the selected column
- **COUNT()** - returns the number of rows in a set
- **SUM()** - returns the total sum of a numerical column
- **AVG()** - returns the average value of a numerical column

SQL Server / MS Access Syntax:

```
SELECT TOP number|percent column_name(s)
FROM table_name
WHERE condition;
```

MySQL Syntax:

```
SELECT column_name(s)
FROM table_name
WHERE condition
LIMIT number;
```

```
SELECT column_name(s)
FROM table_name
WHERE condition
LIMIT number OFFSET SkipRows;
```

```
SELECT column1, column2, ...
FROM table_name
WHERE columnN LIKE pattern;
```

- The percent sign % represents zero, one, or multiple characters
- The underscore sign _ represents one, single character
- a%=> axx
- %a=>xxa
- %a%=>xxaxx
- _a%=>xaxx
- a%0=>axx0
- a__%=>axxxx(minimum 3 character)

SCRIPT

```
use sql_joins;
show tables;
CREATE TABLE TableA (
    A INT
);
INSERT INTO TableA (A) VALUES (1), (1), (1), (0), (0);
```

```

CREATE TABLE TableB (
    B INT
);
INSERT INTO TableB (B) VALUES (1), (0), (1);
select * from tablea;
select * from tableb;

Select * from tablea left join tableb on tablea.A = tableb.B;
Select * from tablea right join tableb on tablea.A = tableb.B;
Select * from tablea inner join tableb on tablea.A = tableb.B;

select * from tablea union select * from tableb;

select * from tablea
group by A;
#1,0

select * from tablea
where tablea.A=tablea.A
group by A;
#1,0

select * from tablea
group by A
order by A ASC;
#0,1

select * from tablea
where tablea.A=tablea.A
group by A
order by A ASC;
#0,1

select * from tablea
order by A ASC;
#0,0,1,1,1

select * from tablea
where tablea.A=tablea.A
group by A
having tablea.A !=0
order by A ASC;
#1

select * from tablea
group by A
having tablea.A !=0;
#1

```

```
select * from tablea
group by A
having tablea.A !=1;
#0

select * from tablea
where exists
(select A from tablea where tablea.A =2);
#zero rows

select * from tablea
where exists
(select A from tablea where tablea.A =0);
#1,1,1,0,0

select count(*) from tablea;
#5

select count(*) from tablea
where tablea.A =1;
#3

select sum(A) from tablea
where tablea.A =1;
#3

select * from tablea limit 0;
#zero rows
select * from tablea limit 1;
#1
select * from tablea limit 5;
#1,1,1,0,0
select A from tablea limit 3 offset 2;
#1,0,0
select A from tablea limit 2 offset 3;
#0,0
select A from tablea limit 1 offset 2;
#1
```

