

# SQLite Interview Questions

## 1) What is SQLite?

SQLite is a **relational database management system** which is **self-contained, server-less** and need **zero configuration**.

## 2) Who was the designer of SQLite?

SQLite was designed by **D. Richard Hipp** for the purpose of no administration required for operating a program.

---

## 3) What are the most important features of SQLite?

There are lots of features which make SQLite very popular:

- SQLite is free of cost.
  - SQLite is server-less.
  - SQLite is flexible.
  - SQLite doesn't need configuration.
  - SQLite is cross-platform.
- 

## 4) What are the advantages of using SQLite?

- SQLite is very light weight database.
  - Data storing is very easy and efficient.
  - SQLite is very easy to learn and use.
- 

## 5) How would you create a database in SQLite?

In SQLite, `sqlite3` command is used to create database.

**Syntax:**

### 6) How would you create a table in SQLite database?

CREATE TABLE statement is used to create a table in SQLite database. You have to define the columns and data types of each column while creating the table.

#### Syntax:

```
CREATE TABLE database_name.table_name(  
column1 datatype PRIMARY KEY(one or more columns),  
column2 datatype,  
column3 datatype,  
columnN datatype,  
);
```

---

### 7) How would you drop a table in SQLite database?

**DROP TABLE** command is used to delete or **permanently drop** a table from **SQLite database**.

#### Syntax:

```
DROP TABLE table_name;
```

---

### 8) How would you create an AUTOINCREMENT field?

For autoincrement, you have to declare a column of the table to be INTEGER PRIMARY KEY, then whenever you insert a NULL into that column of the table, the NULL is automatically converted into an integer which is one greater than the largest value of that column over all other rows in the table, or 1 if the table is empty.

---

### 9) What data types are supported by SQLite?

SQLite uses dynamic typing. Content can be stored as **INTEGER, REAL, TEXT, BLOB**, or as **NULL**.

---

## 10) How to insert data in a table in SQLite?

**INSERT INTO** statement is used to insert data in a table in SQLite database. There are two ways to insert data in table:

### Method1:

#### Syntax:

```
INSERT INTO TABLE_NAME [(column1, column2, column3,...columnN)]  
VALUES (value1, value2, value3,...valueN);
```

### Method2:

#### Syntax:

```
INSERT INTO TABLE_NAME VALUES (value1,value2,value3,...valueN);
```

---

## 11) How would you retrieve data from SQLite table?

The **SELECT** command is used to retrieve data from SQLite table. If you want to retrieve all columns from the table use **SELECT \*** otherwise use the specific column's name separated by commas.

#### Syntax:

```
SELECT * FROM table_name;  
Or  
SELECT column1, column2, columnN FROM table_name;
```

---

## 12) What is the use of UPADTE query in SQLite?

The **UPDATE** query is used to modify the existing records in the SQLite table. You have to use the **WHERE clause** to modify a specific row otherwise all rows will be updated.

#### Syntax:

```
UPDATE table_name  
SET column1 = value1, column2 = value2...., columnN = valueN  
WHERE [condition];
```

---

### 13) How can you delete the existing records from a table in SQLite?

In SQLite, **DELETE** command is used to delete the existing records from a table. You should use the **WHERE clause** to choose the specific row otherwise all rows will be deleted.

**Syntax:**

```
DELETE FROM table_name  
WHERE [conditions.....];
```

---

### 14) What is the use of WHERE clause in CRUD statements?

**WHERE clause** is used to refer a specific row where the CRUD operation is executed. Without using WHERE clause all the rows will be affected.

---

### 15) What is the usage of AND & OR operators with WHERE clause?

**AND** & **OR** operators are used with **WHERE clause** to combine two or more than two conditions together.

**Syntax:**

```
SELECT column1, column2, columnN  
FROM table_name  
WHERE [condition1] AND [condition2]...OR [conditionN];
```

---

### 16) What is the usage of LIKE operator with WHERE clause?

The **LIKE** operator is used to match text values against a pattern using wildcards. It uses two wildcards **%** and **\_** with string for matching with input.

**Syntax:**

```
SELECT FROM table_name  
WHERE column LIKE 'XXXX%'  
  
SELECT FROM table_name  
WHERE column LIKE 'XXXX_'
```

---

## 17) What is the use of LIMIT clause with SELECT query?

**LIMIT** clause is used with **SELECT** statement when we want a limited number of fetched records.

**Syntax:**

```
SELECT column1, column2, columnN  
FROM table_name  
LIMIT [no of rows]
```

---

## 18) Why is ORDER BY clause used with SELECT statement?

The ORDER BY clause is used to sort the fetched data in a specific order either ascending or descending.

**Syntax:**

```
SELECT column-list  
FROM table_name  
[WHERE condition]  
[ORDER BY column1, column2, .. columnN] [ASC | DESC];
```

---

## 19) What is the use of SQLite GROUP BY clause?

SQLite GROUP BY clause is used to collect the same elements into a group. It is used with SELECT statement.

**Syntax:**

```
SELECT column-list  
FROM table_name  
WHERE [ conditions ]  
GROUP BY column1, column2....columnN  
ORDER BY column1, column2....columnN
```

Note: you can use GROUP BY and ORDER BY clauses together.

---

## 20) What is the use of DISTINCT clause in SQLite?

The **DISTINCT** clause is always used with **SELECT** statement. It is used to retrieve only unique records and restrict the duplicate entries.

It is used when the table has multiple duplicate records.

**Syntax:**

```
SELECT DISTINCT column1, column2,.....columnN  
FROM table_name  
WHERE [condition]
```

---

## 21) What is UNION operator? How does it work?

SQLite **UNION** Operator is used to combine the result set of two or more tables using **SELECT** statement. Both the tables must have same number of fields in result table.

**Syntax:**

```
SELECT expression1, expression2, ... expression_n  
FROM tables  
[WHERE conditions]  
UNION  
SELECT expression1, expression2, ... expression_n  
FROM tables  
[WHERE conditions];
```

---

## 22) What is UNION ALL operator? What is the difference between UNION and UNION ALL operator?

The UNION ALL operator is used to combine the result of two or more tables using SELECT statement. The unique difference between UNION and UNION ALL operator is that UNION operator ignores the duplicate entries while combining the results while UNION ALL doesn't ignore duplicate values.

**Syntax:**

```
SELECT expression1, expression2, ... expression_n  
FROM tables  
[WHERE conditions]  
UNION ALL
```

```
SELECT expression1, expression2, ... expression_n
FROM tables
[WHERE conditions];
```

---

## 23) What is SQLite JOIN? How many types of JOINS are supported in SQLite?

SQLite **JOIN clause** is used to combine two or more tables in a database. It combines the table by using the common values of the both table.

There are mainly three types of JOINS supported in SQLite:

- SQLite INNER JOIN
  - SQLite OUTER JOIN
  - SQLite CROSS JOIN
- 

## 24) What is SQLite INNER JOIN?

SQLite **INNER JOIN** is simplest and most common join. It combines all rows from both tables where the condition is satisfied.

**Syntax:**

```
SELECT ... FROM table1 [INNER] JOIN table2 ON conditional_expression ...
```

---

## 25) What is SQLite OUTER JOIN?

There are three types of OUTER JOINS:

- Left outer join
- Right outer join
- Full outer join

But SQLite only supports left outer join. The SQLite left outer join returns all rows from left table and only those rows from the right table where the join condition is satisfied.

**Syntax:**

```
SELECT ... FROM table1 LEFT OUTER JOIN table2 ON conditional_expression
```

---

## 26) Explain SQLite CROSS JOIN.

The SQLite Cross join is used to match every rows of the first table with every rows of the second table. If the first table contains x columns and second table contains y columns then the resultant Cross join table will contain the  $x*y$  columns.

### Syntax:

```
SELECT ... FROM table1 CROSS JOIN table2
```

---

## 27) What is SQLite date and time () function?

SQLite date and time () functions are used to retrieve current date and time and also do calculations on the dates and time.

There are mainly 6 types of date and time () function in SQLite:

- SQLite date() Function
  - SQLite datetime() Function
  - SQLite julianday() Function
  - SQLite now() Function
  - SQLite strftime() Function
  - SQLite time() Function
- 

## 28) What is the use of date() function in SQLite?

The **SQLite date()** function is used to fetch the date and show it in 'YYYY-MM-DD' format.

### Syntax:

```
date(timestring, [ modifier1, modifier2, ... modifier_n ] )
```

---

## 29) What is the use of datetime() function in SQLite?

The SQLite **datetime()** function is used to retrieve current date and time in 'YYYY-MM-DD HH:MM:SS' format.



### Syntax:

**datetime(timestring, [ modifier1, modifier2, ... modifier\_n ] )**

---

## 30) What is SQLite julianday() function?

A Julian Day is the number of days since Nov 24, 4714 BC 12:00pm Greenwich time in the **Gregorian** calendar. So, the **julianday()** function is used to return number of days **since Nov 24, 4714 BC 12:00pm**.

### Syntax:

**julianday(timestring [, modifier1, modifier2, ... modifier\_n ] )**

---

## 31) What is the use of SQLite now() function?

SQLite now is not a function. Instead of this, it is a timestring parameter which is used to fetch current date and time.

### Syntax:

**date('now')**

or

**time('now')**

---

## 32) What is the usage of SQLite strftime() function?

SQLite **strftime()** function is used to fetch date and time and also perform time and date calculation.

### Syntax:

**strftime(format, timestring [, modifier1, modifier2, ... modifier\_n ] )**

---

## 33) What is the use of SQLite time() function?

SQLite **time()** function is used to fetch current time in **'HH-MM-SS'** format.

### Syntax:

**time(timestring, [ modifier1, modifier2, ... modifier\_n ] )**

---

### 34) What do you understand by SQLite aggregate functions?

SQLite **aggregate** functions are the type of functions where values of multiple rows are grouped together as input on certain criteria and form a single value as output.

There are many types of aggregate functions in SQLite.

---

### 35) What is SQLite MIN aggregate function?

SQLite **MIN** aggregate function is used to retrieve the minimum value of the expression.

**Syntax:**

```
SELECT MIN(aggregate_expression)
FROM tables
[WHERE conditions];
```

---

### 36) What is SQLite MAX aggregate function?

SQLite MAX aggregate function is used to fetch the maximum value of an expression.

**Syntax:**

```
SELECT MAX(aggregate_expression)
FROM tables
[WHERE conditions];
```

---

### 37) What is SQLite AVG aggregate function?

The SQLite AVG function returns the average value of the expression.

**Syntax:**

```
SELECT AVG(aggregate_expression)
FROM tables
[WHERE conditions];
```

---

### 38) What is SQLite COUNT aggregate function?

The SQLite COUNT function is used to retrieve total count of an expression.

**Syntax:**

```
SELECT COUNT(aggregate_expression)
FROM tables
[WHERE conditions];
```

---

### 39) What is SQLite SUM aggregate function?

The SQLite SUM aggregate function is used to get the total summed value of an expression.

**Syntax:**

```
SELECT SUM(aggregate_expression)
FROM tables
[WHERE conditions];
```

---

### 40) What is the difference between SQL and SQLite?

The main differences between SQL and SQLite are:

- SQL is Structured Query Language while SQLite is a relational database management
  - system mostly used in android mobile devices to store data.
  - SQL support stored procedures while SQLite does not support stored procedures.
  - SQL is server based while SQLite is file based.
- 

### 41) What is SQLite Transactions?

Transaction specifies a unit of work that is performed against a database. The transaction's

properties are determined by ACID:

**Atomicity:** Atomicity is a property which specifies that all work units are successfully completed.

**Consistency:** It is used to ensure that the database changes states upon a successfully committed transaction.

**Isolation:** It facilitates you to operate transaction independently of and transparent to each other.

**Durability:** It ensures that the result or effect of a committed transaction persists in case of a

system failure.

---

## 42) In which areas, SQLite is preferred most?

SQLite is preferred to work with:

- Embedded devices.
  - Application file format.
  - Data Analysis.
  - Websites.
  - File archives.
  - Cache for enterprise data.
  - Server side database.
  - Internal or temporary databases.
  - Replacement for ad hoc disk files.
  - Experimental SQL language extensions.
- 

## 43) What is the use of .dump command in SQLite?

The **.dump** command is used to make a SQLite database dump. Once you use the dump command all your data will be dumped forever and cannot be retrieved.