

# SQLITE - INDEXED BY

[http://www.tutorialspoint.com/sqlite/sqlite\\_indexed\\_by.htm](http://www.tutorialspoint.com/sqlite/sqlite_indexed_by.htm)

Copyright © tutorialspoint.com

The "INDEXED BY index-name" clause specifies that the named index must be used in order to look up values on the preceding table.

If index-name does not exist or cannot be used for the query, then the preparation of the SQLite statement fails.

The "NOT INDEXED" clause specifies that no index shall be used when accessing the preceding table, including implied indices create by UNIQUE and PRIMARY KEY constraints.

However, the INTEGER PRIMARY KEY can still be used to look up entries even when "NOT INDEXED" is specified.

## Syntax

Following is the syntax for INDEXED BY clause and it can be used with DELETE, UPDATE or SELECT statement:

```
SELECT|DELETE|UPDATE column1, column2...  
INDEXED BY (index_name)  
table_name  
WHERE (CONDITION);
```

## Example

Consider table [COMPANY](#) we will create an index and use it for performing INDEXED BY operation.

```
sqlite> CREATE INDEX salary_index ON COMPANY(salary);  
sqlite>
```

Now selecting the data from table COMPANY you can use INDEXED BY clause as follows:

```
sqlite> SELECT * FROM COMPANY INDEXED BY salary_index WHERE salary > 5000;
```

This would produce the following result:

ID	NAME	AGE	ADDRESS	SALARY
7	James	24	Houston	10000.0
2	Allen	25	Texas	15000.0
1	Paul	32	California	20000.0
3	Teddy	23	Norway	20000.0
6	Kim	22	South-Hall	45000.0
4	Mark	25	Rich-Mond	65000.0
5	David	27	Texas	85000.0