Advanced Database Design
Sorting and Indexing

Gregory S. DeLozier, Ph.D.
gdelozie@kent.edu

This Week's Topics

- No quiz today
- Do the todo list/ORM together
- Publish the code on the web
- Looking at query optimization
- A query optimization experiment

If we have time…

- An introduction to NoSQL
- A tour of a simple Mongo application
- Homework

Do the todo list/ORM together

Demo time!

Publish the code on the Web

- Make an account at pythonanywhere.com
- Learn how to create a simple web app
  - Slightly modified procedures
- How to get files to PA
- Bringing up our application

Demo time!

Looking at Query Optimization

- Tables vs Indices
- A table is a set of records
- An index is a tree (usually) that can be searched
  - For a single field
  - For multiple fields
- The tree result is an index (i.e. row number)
- It matters a great deal if indices are created
- Good examples online for SQLIte

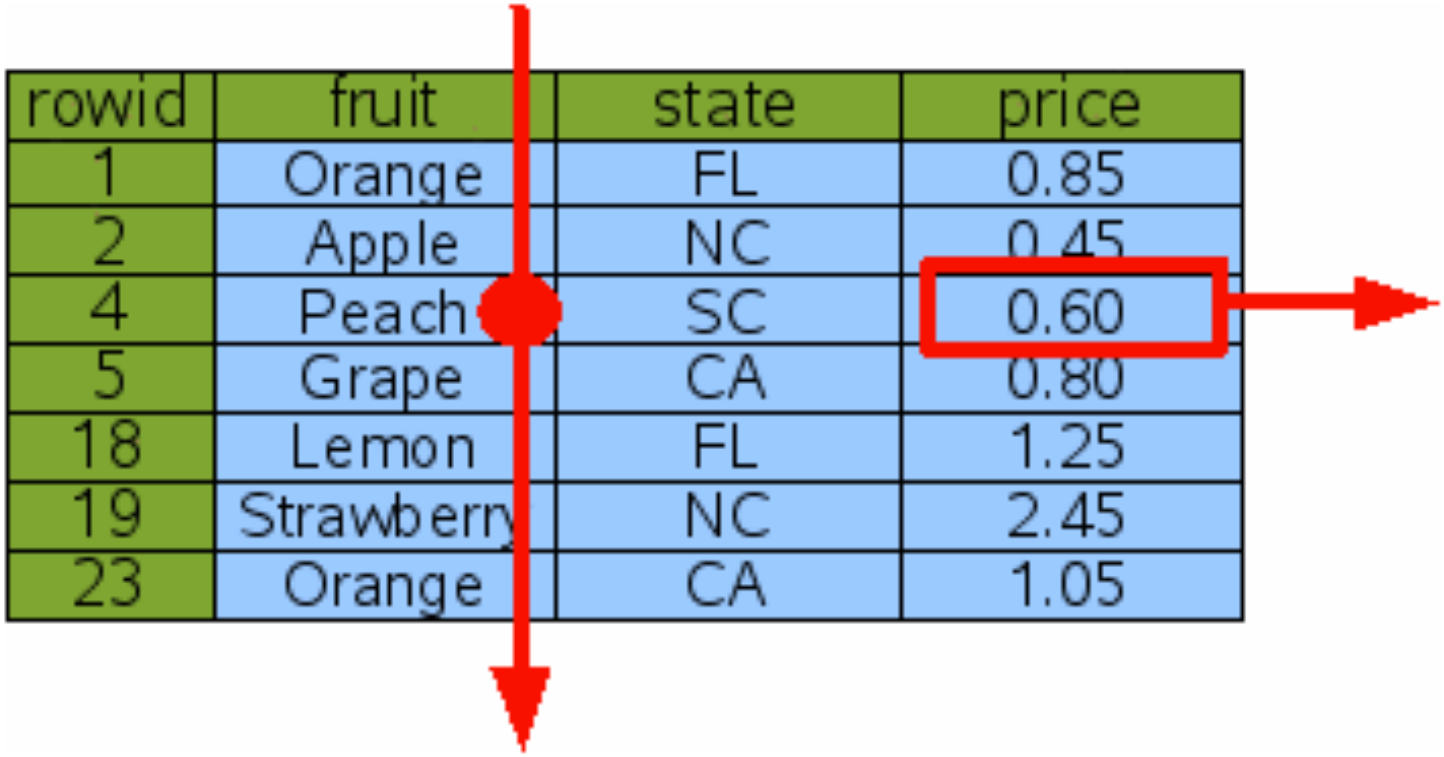
Optimizer Strategies

rowid	fruit	state	price
1	Orange	FL	0.85
2	Apple	NC	0.45
4	Peach	SC	0.60
5	Grape	CA	0.80
18	Lemon	FL	1.25
19	Strawberry	NC	2.45
23	Orange	CA	1.05

Figure 1: Logical Layout Of Table "FruitsForSale"

Full Table Scan

```
SELECT price FROM fruitsforsale WHERE fruit='Peach';
```

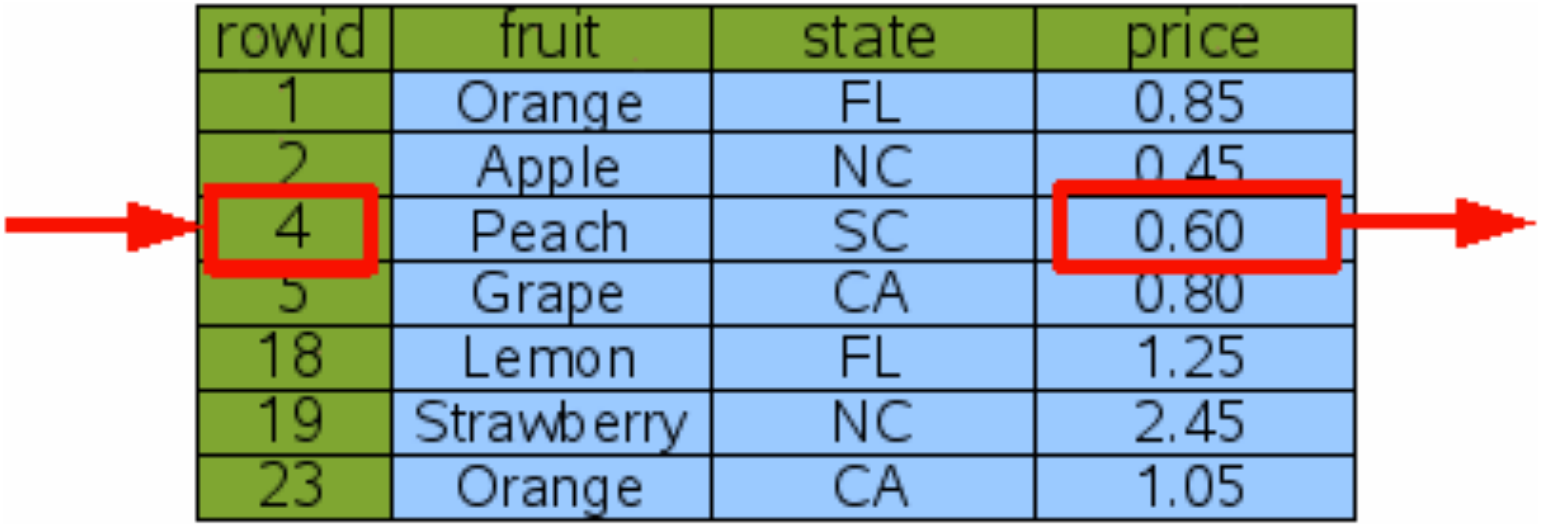


rowid	fruit	state	price
1	Orange	FL	0.85
2	Apple	NC	0.45
4	Peach	SC	0.60
5	Grape	CA	0.80
18	Lemon	FL	1.25
19	Strawberry	NC	2.45
23	Orange	CA	1.05

Figure 2: Full Table Scan

Row Number Fetch

```
SELECT price FROM fruitsforsale WHERE rowid=4;
```



rowid	fruit	state	price
1	Orange	FL	0.85
2	Apple	NC	0.45
4	Peach	SC	0.60
5	Grape	CA	0.80
18	Lemon	FL	1.25
19	Strawberry	NC	2.45
23	Orange	CA	1.05

Figure 3: Lookup By Rowid

Create an Index

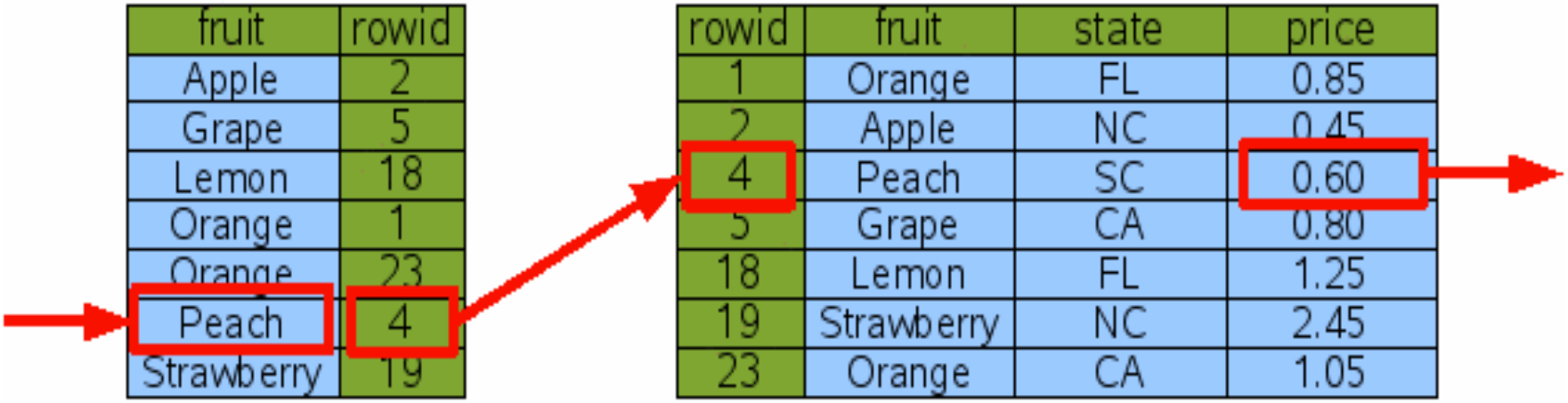
```
CREATE INDEX idx1 ON fruitsforsale(fruit);
```

fruit	rowid
Apple	2
Grape	5
Lemon	18
Orange	1
Orange	23
Peach	4
Strawberry	19

Figure 4: An Index On The Fruit Column

Using an Index

```
SELECT price FROM fruitsforsale WHERE fruit='Peach';
```



fruit	rowid
Apple	2
Grape	5
Lemon	18
Orange	1
Orange	23
Peach	4
Strawberry	19

rowid	fruit	state	price
1	Orange	FL	0.85
2	Apple	NC	0.45
4	Peach	SC	0.60
5	Grape	CA	0.80
18	Lemon	FL	1.25
19	Strawberry	NC	2.45
23	Orange	CA	1.05

Figure 5: Indexed Lookup For The Price Of Peaches

Using an Index

```
SELECT price FROM fruitsforsale WHERE fruit='Orange'
```

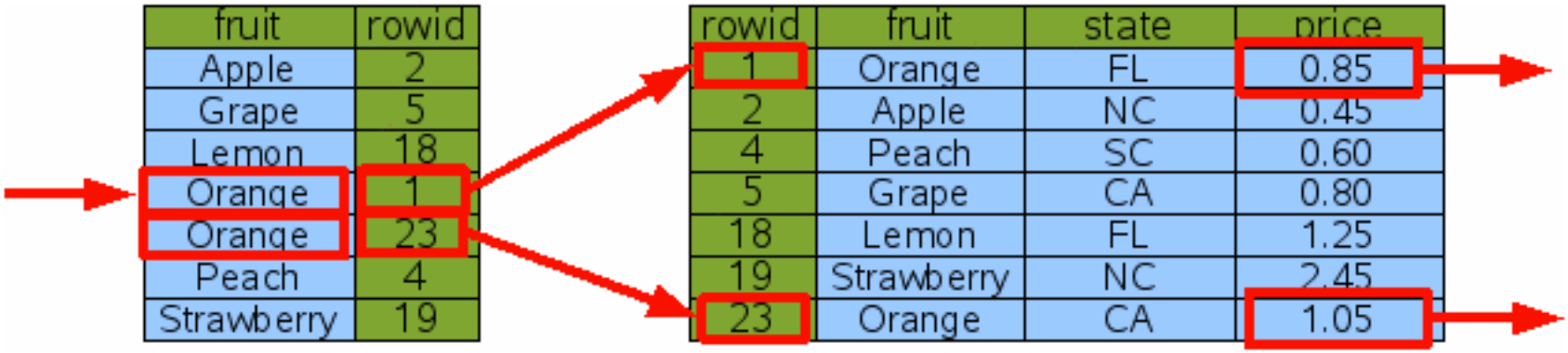


Figure 6: Indexed Lookup For The Price Of Oranges

Using an Index

```
SELECT price FROM fruitsforsale WHERE fruit='Orange' AND state='CA'
```

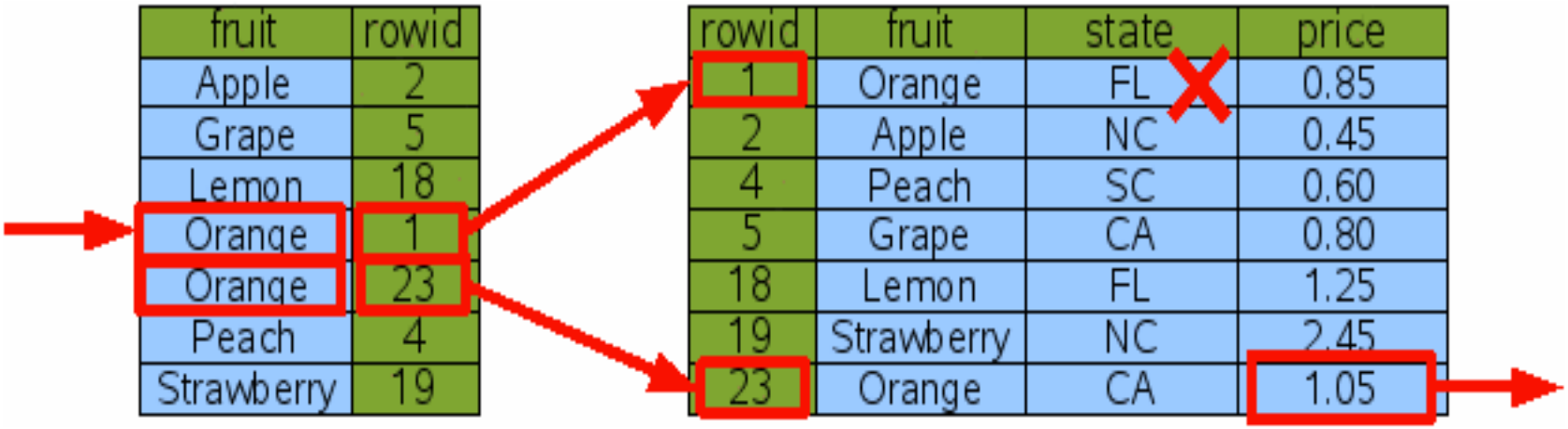


Figure 7: Indexed Lookup Of California Oranges

Creating an Alternate Index

```
CREATE INDEX Idx2 ON fruitsforsale(state);
```

state	rowid
CA	5
CA	23
FL	1
FL	18
NC	2
NC	19
SC	4

Figure 8: Index On The State Column

Using an Alternate Index

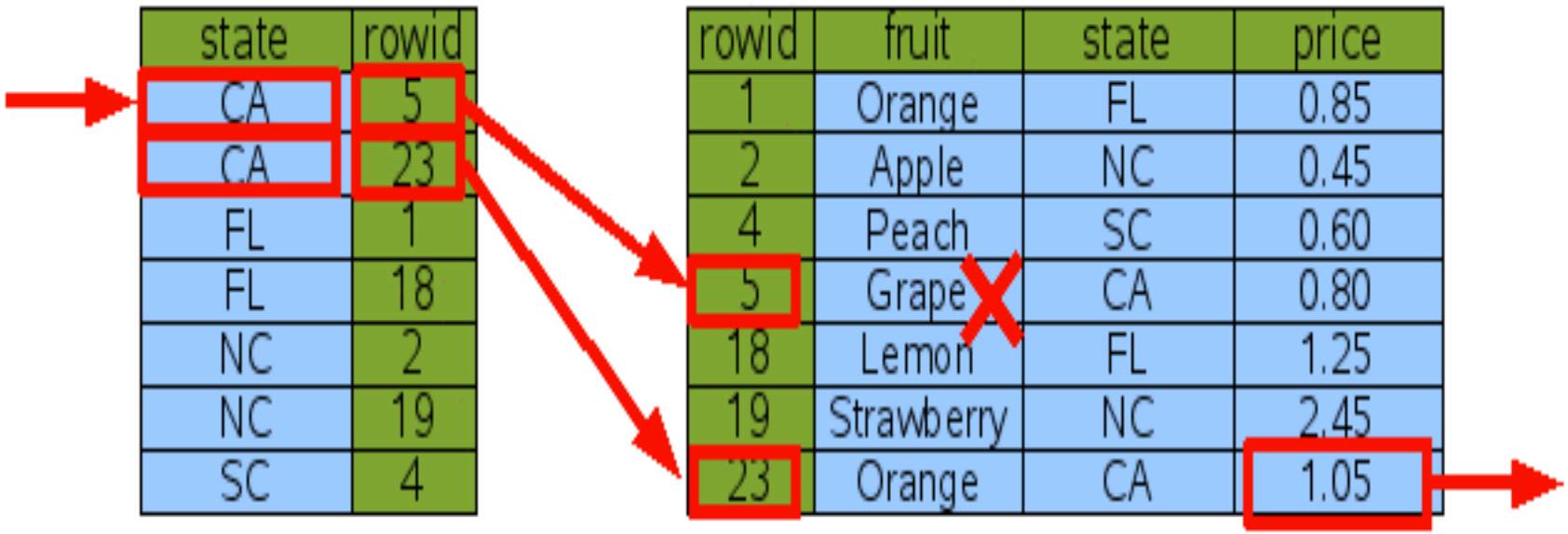


Figure 9: Indexed Lookup Of California Oranges

Creating a Multiple Column Index

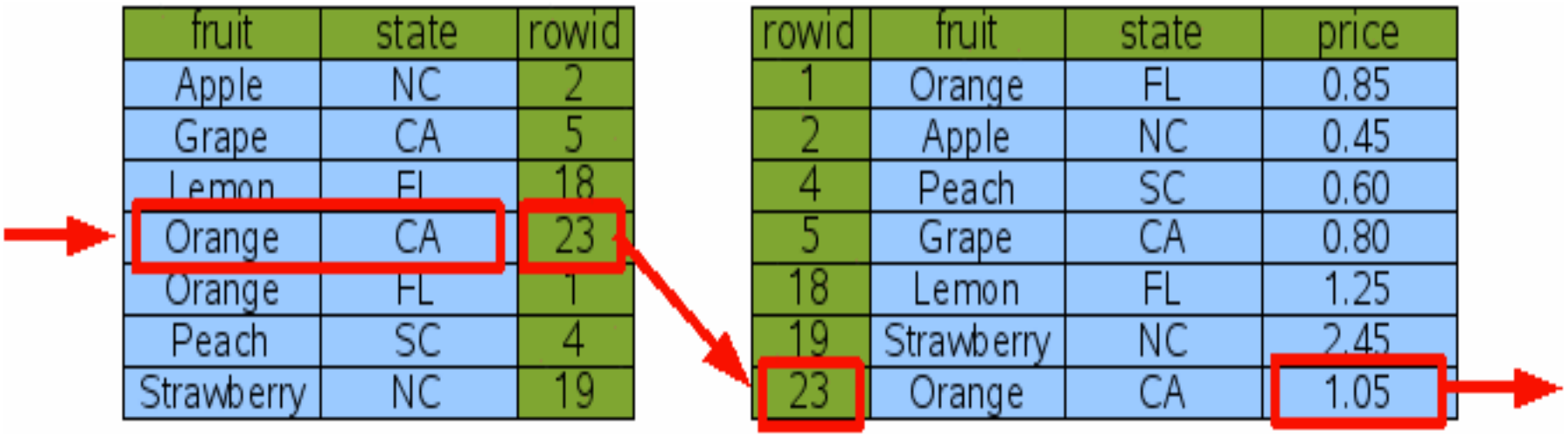
```
CREATE INDEX Idx3 ON FruitsForSale(fruit, state);
```

fruit	state	rowid
Apple	NC	2
Grape	CA	5
Lemon	FL	18
Orange	CA	23
Orange	FL	1
Peach	SC	4
Strawberry	NC	19

Figure 1: A Two-Column Index

Using a Multiple Column Index

```
SELECT price FROM fruitsforsale WHERE fruit='Orange' AND state='CA'
```



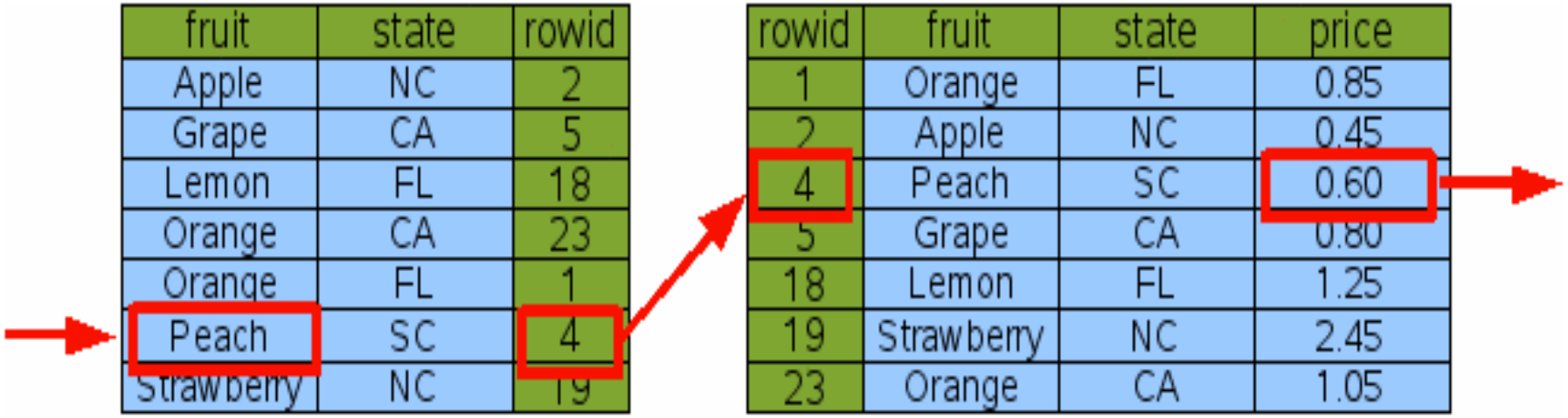
fruit	state	rowid
Apple	NC	2
Grape	CA	5
Lemon	FL	18
Orange	CA	23
Orange	FL	1
Peach	SC	4
Strawberry	NC	19

rowid	fruit	state	price
1	Orange	FL	0.85
2	Apple	NC	0.45
4	Peach	SC	0.60
5	Grape	CA	0.80
18	Lemon	FL	1.25
19	Strawberry	NC	2.45
23	Orange	CA	1.05

Figure 11: Lookup Using A Two-Column Index

Using a Multiple Column Index

```
SELECT price FROM fruitsforsale WHERE fruit='Peach'
```



fruit	state	rowid
Apple	NC	2
Grape	CA	5
Lemon	FL	18
Orange	CA	23
Orange	FL	1
Peach	SC	4
Strawberry	NC	19

rowid	fruit	state	price
1	Orange	FL	0.85
2	Apple	NC	0.45
4	Peach	SC	0.60
5	Grape	CA	0.80
18	Lemon	FL	1.25
19	Strawberry	NC	2.45
23	Orange	CA	1.05

Figure 12: Single-Column Lookup On A Multi-Column Index

Using Index Values

- Where clause can search using:

```
column = expression
column > expression
column >= expression
column < expression
column <= expression
expression = column
expression > column
expression >= column
expression < column
expression <= column
column IN (expression-list)
column IN (subquery)
column IS NULL
```

- > create inde*x* *ta*ble(a,b…)

Creating a Covering Index

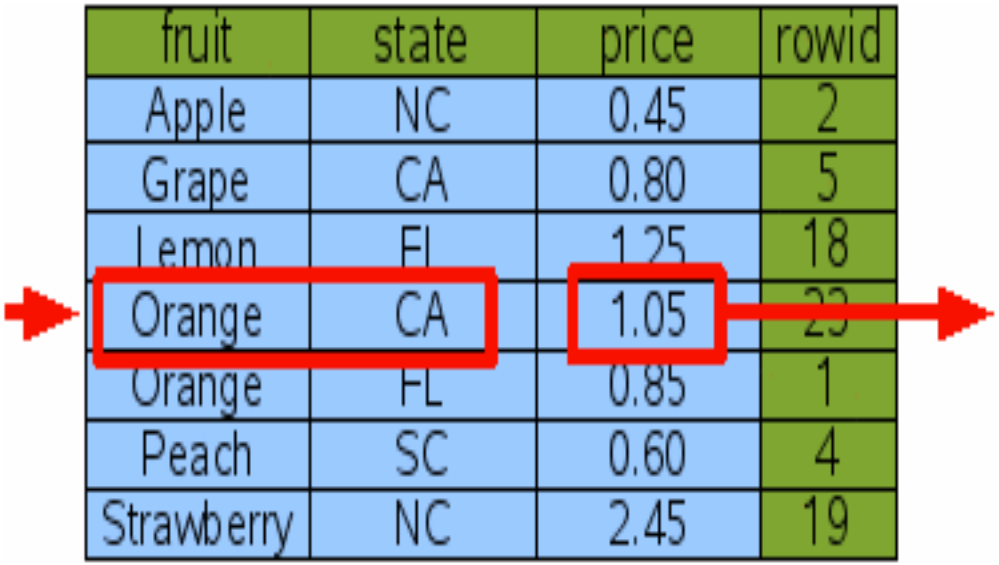
```
CREATE INDEX Idx4 ON FruitsForSale(fruit, state, price);
```

fruit	state	price	rowid
Apple	NC	0.45	2
Grape	CA	0.80	5
Lemon	FL	1.25	18
Orange	CA	1.05	23
Orange	FL	0.85	1
Peach	SC	0.60	4
Strawberry	NC	2.45	19

Figure 13: A Covering Index

Using a Covering Index

```
SELECT price FROM fruitsforsale WHERE fruit='Orange' AND state='CA';
```



fruit	state	price	rowid
Apple	NC	0.45	2
Grape	CA	0.80	5
Lemon	FL	1.25	18
Orange	CA	1.05	23
Orange	FL	0.85	1
Peach	SC	0.60	4
Strawberry	NC	2.45	19

Figure 14: Query Using A Covering Index

Using an OR search

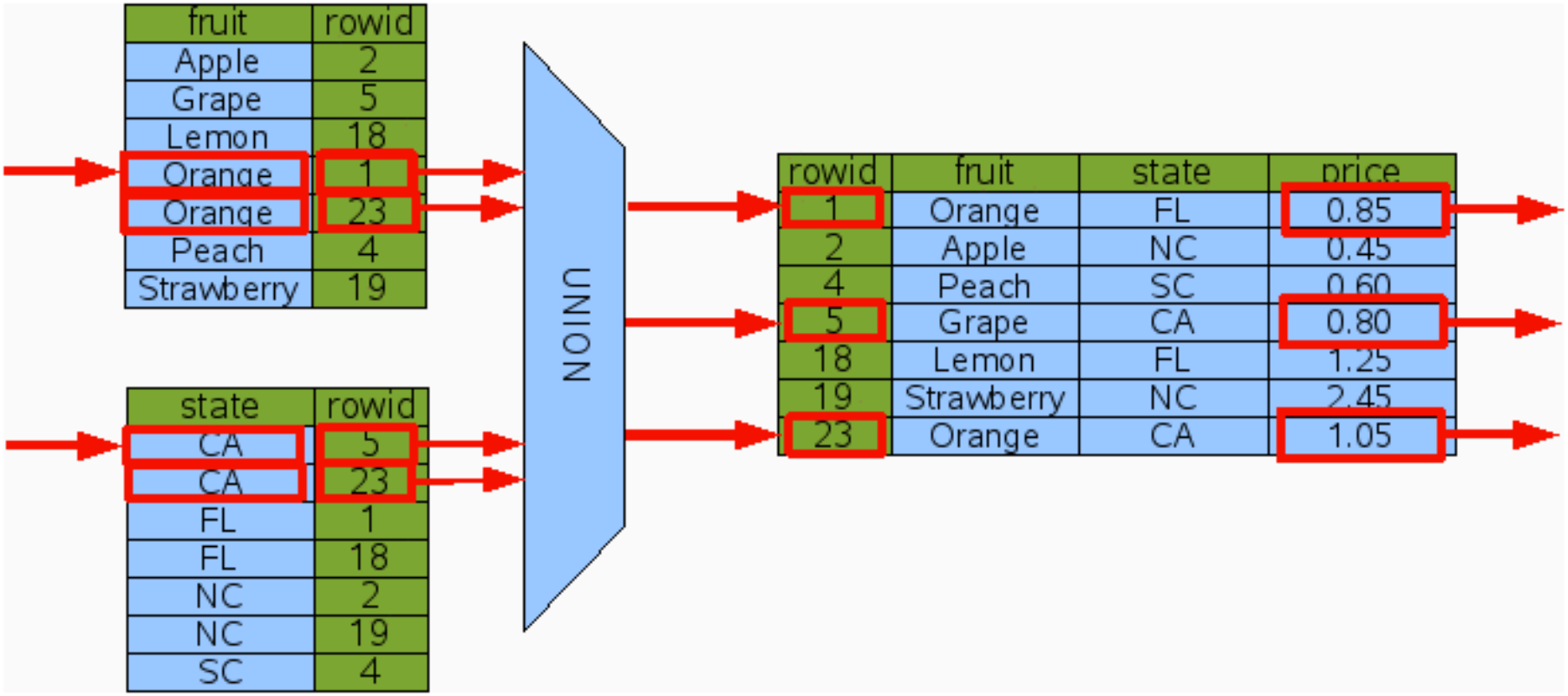


Figure 15: Query With OR Constraints

Sorting

```
SELECT * FROM fruitsforsale ORDER BY fruit;
```

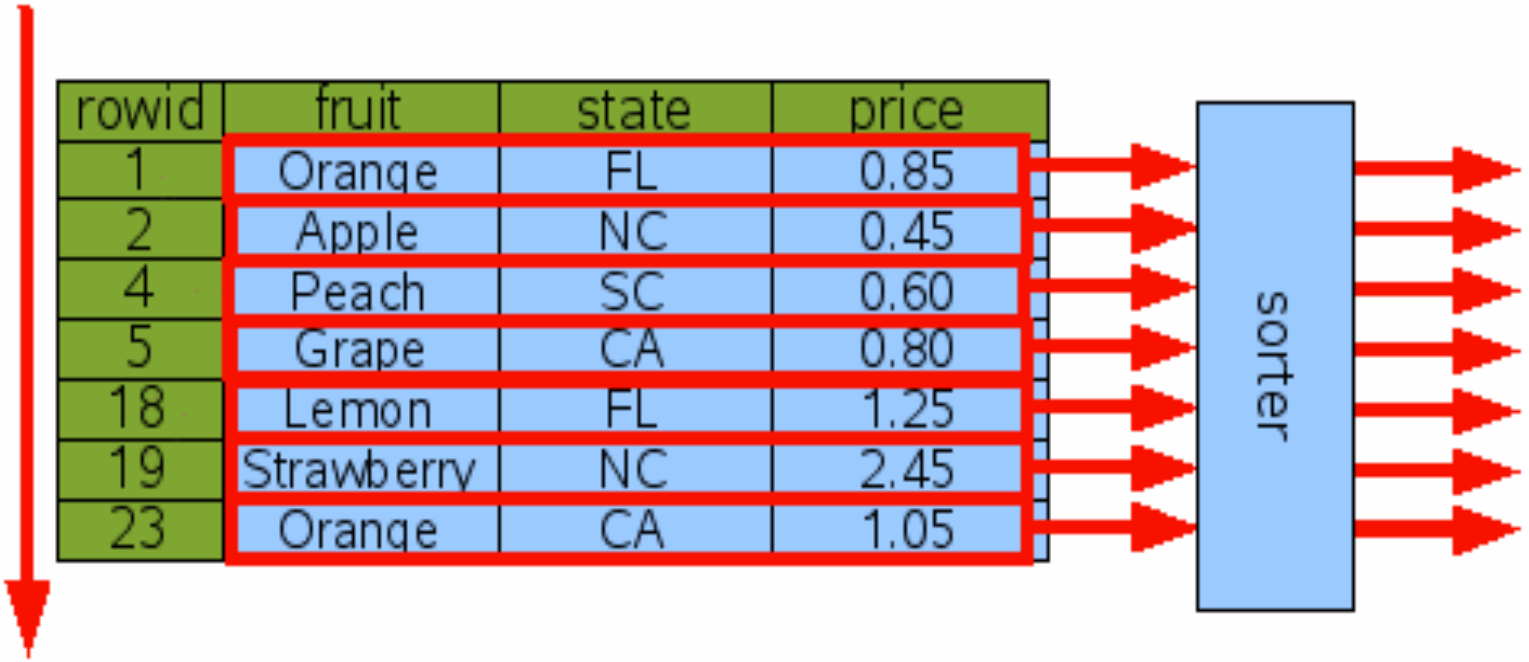
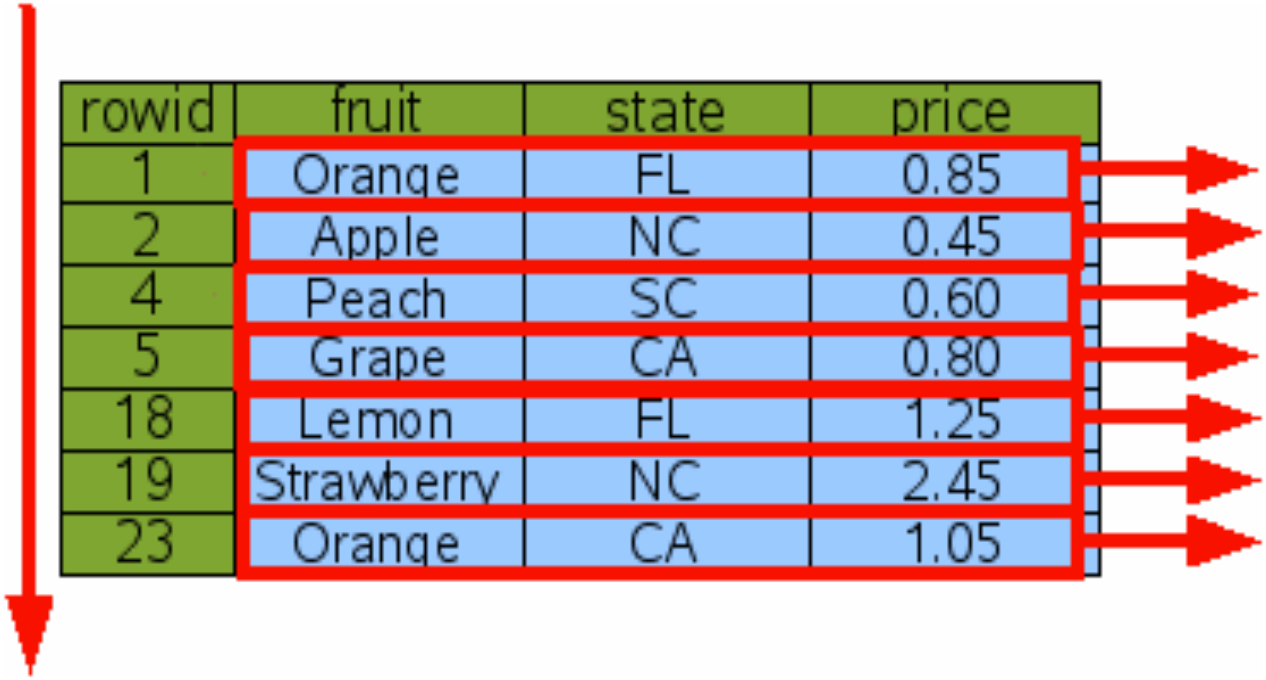


Figure 16: Sorting Without An Index

Sorting

```
SELECT * FROM fruitsforsale ORDER BY rowid;
```



rowid	fruit	state	price
1	Orange	FL	0.85
2	Apple	NC	0.45
4	Peach	SC	0.60
5	Grape	CA	0.80
18	Lemon	FL	1.25
19	Strawberry	NC	2.45
23	Orange	CA	1.05

Figure 17: Sorting By Rowid

Sorting

```
SELECT * FROM fruitsforsale ORDER BY fruit;
```

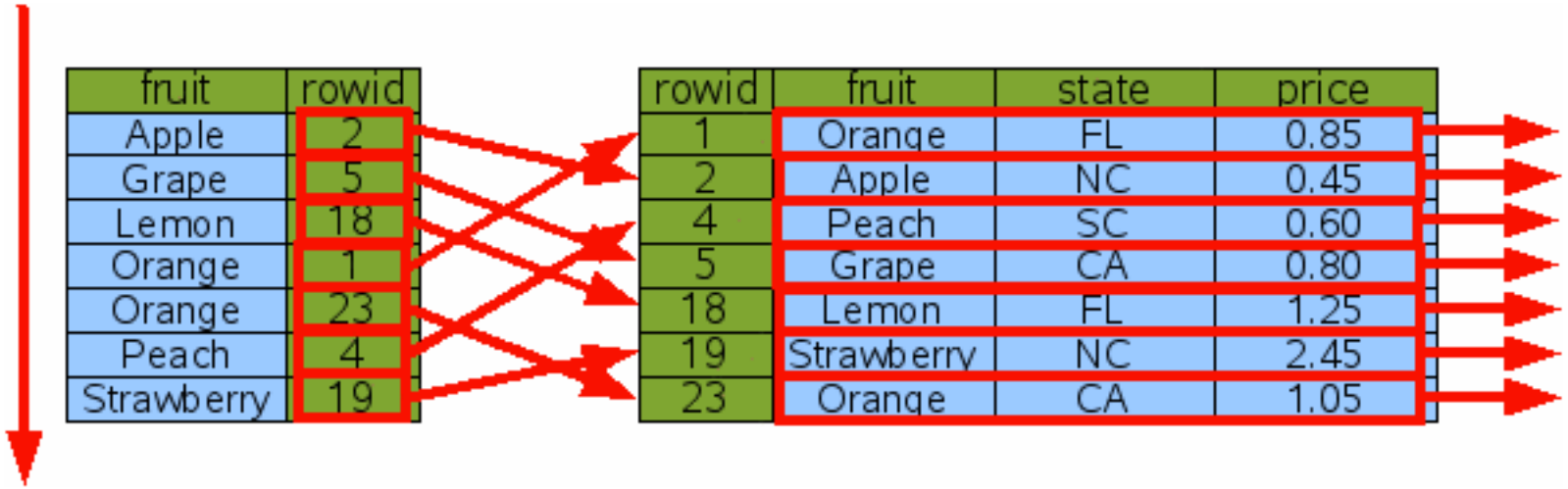


Figure 18: Sorting With An Index

Sorting

Figure 19: Sorting With A Covering Index

Partial Sorting

```
SELECT * FROM fruitforsale ORDER BY fruit, price
```

Figure 22: Partial Sort By Index

Optimizer Strategies

- SQLite uses query strategies
  - http://www.sqlite.org/queryplanner.html
- SQLite planner does optimization
  - Uses strategies to decide how to search
  - ttps://www.sqlite.org/optoverview.html
- Query plans come with explanations
  - http://www.sqlite.org/eqp.html