Indexing in SQLite

SQLite supports B+ tree index. By default only *unique* and *primary* key columns are indexed

Example

```
CREATE TABLE book(
        callno char(10),
        author char(20),
        title char(30),
        year char(4),
        PRIMARY KEY (callno)
);
```

Create an index on author

```
CREATE INDEX authidx ON book (author)
```

This creates an unclustered (dense) index on author.

But we can also make it UNIQUE

```
CREATE UNIQUE INDEX authidx ON book (author)
```

Making it clustered on primary key

```
CREATE TABLE book(
        callno char(10),
        author char(20),
        title char(30),
        year char(4),
        PRIMARY KEY (callno)
) WITHOUT ROWID
```

Primary key is required for **without rowid** tables

Indexing in Oracle

Creating an un-clustered index

```
CREATE TABLE book(
        callno char(10),
        author char(20),
        title char(30),
        year char(4),
        PRIMARY KEY (callno)
);
```

```
CREATE INDEX authidx ON book (author);
```

Results: an **un-clustered dense** index on author

Creating a clustered index on primary key

```
CREATE TABLE book(
        callno char(10),
        author char(20),
        title char(30),
        year char(4),
        PRIMARY KEY (callno)
) ORGANIZATION INDEX;
```

This syntax allows a clustered index on the primary key of the table only.

Creating clustered index on non-primary key

```
CREATE TABLE book(
        callno char(10),
        author char(20),
        title char(30),
        year char(4),
        PRIMARY KEY (callno)
) cluster authcl(author);
```

```
CREATE INDEX authidx on cluster authcl;
```

An Oracle cluster may contain rows from more than one table

Indexing in DB2

Create un-clustered indexes on callno and author

```
CREATE TABLE book(
        callno char(10),
        author char(20),
        title char(30),
        year char(4),
        PRIMARY KEY (callno)
);
```

```
CREATE INDEX callno_idx on book (callno);
CREATE INDEX auth_idx on book (author);
```

Can make (only) one index clustered

```
CREATE INDEX auth_idx on book (author) cluster;
```

Data must be preferable sorted on clustering column(s) in the OS file

Choosing an Index

Example 1

```
SELECT E. Id
FROM Employee E
WHERE E.Salary < :upper AND E.Salary > :lower
```

- a range search on Salary
- suppose the primary key is employee id
    - it is likely that there is a main clustered index on that attribute that is of no use for this query
- Choose a secondary B+ tree index with search key **salary**

Example 2

```
SELECT T.StudId
FROM Transcript T
WHERE T.Grade = :grade
```

- an equality search on Grade
- Suppose the primary key is (StudId, Semester, CrsCode)
    - it is likely that there is a main clustered index on these attributes that is for no use for this query
- Choose a secondary B+ tree or hash index with search key **Grade**

Example 3

```
SELECT T.CrsCode, T.Grade
FROM Transcript T
WHERE T.StudId = :id AND T.Semester = ‘F2000’
```

- equality search on StudId and Semester
- If the primary key is (StudeId, Semester, CrsCode) it is likely that there is a main clustered index on this sequence of attributes
- If the main index is a B+ tree it can be used for this search
- If the main index is a hash, it cannot be used for this search. Choose B+ tree or hash with search key StudId or (StudId, Semester)