Data Modeling and Databases: Project Phase 3

Due on Sunday, November 22, 2015

Qiang Qu

Alexey Chernyshov, Vladislav Kravchenko, Murad Magomedov

Contents

1 Phase 3. DBMS

1.1 Introducing

According to Project Phase 3 requirements we had to replace the relational database with our own implementation of a database. So we have done it, including implementation of query operators using python programming language.

1.2 Functionality

At phase 3 all SQL code was replaced with calls to the query operators. So now our database provides methods to query data, to insert new records and to update existing records.

For example, SQL code of Select:

```
cur.execute("""SELECT id, name, institute FROM author WHERE id=%s;""", (id, ))
```

Replaced with:

```
qr_author = qp.getFromTable('author', )
qr_res = qp.getFromTable('author', ('id', id))
qr_res = qr_res.project('id', 'name', 'institute')
qr_res = qr_res.sort('id')
```

Delete:

```
cur.execute("""DELETE FROM article_author WHERE author_id=%s""", (id, ))
```

Replaced with:

```
qp.deleteFromTable('article_author', ('author_id', author_id))
```

Update:

```
cur.execute("""UPDATE author SET id=%s, name=%s, institute=%s
            WHERE id=%s""", (id, name, institute, id))
```

Replaced with:

```
qp.deleteFromTable('author', ('id', id))
qp.addToTable('author', ('id', id), ('name', name), ('institute', institute))
```

Insert:

```
cur.execute("""INSERT INTO reference (from_id, to_id)
            VALUES (%s, %s)""", (id, article_id))
```

Replaced with:

```
qp.addToTable('reference', ('from_id', id), ('to_id', article_id))
```

Group by:

```
qres.groupBy('id', 'name')
```

Our database also answers queries that join two tables, for example join of tables article_author and article:

Join:

```
qr_article_author = qp.getFromTable('article_author', ('author_id', author_id))
qr_article = qp.getFromTable('article')
qr_res = qr_article_author.join(qr_article, 'article_id', 'id')
qr_res = qr_res.project('article_id', 'paper_title')
qr_res = qr_res.sort('article_id')
```

One of the requirements of Phase 3 was to use the iterator model to implement query operators, so that operators should pull tuples from underlying operators using next() calls.
So, in our case, class qres (query result) is iterable. It is possible to use qres.next or next(qres) to iterate through results. In our DBMS it is possible to index on Primary keys with the ability to add index on other attributes.
And according to Project limitations we are using a single file for entire DBMS. The Database file is created using psqlloader.py that imports data from Postgres.

1.3 Disk Page Organisation

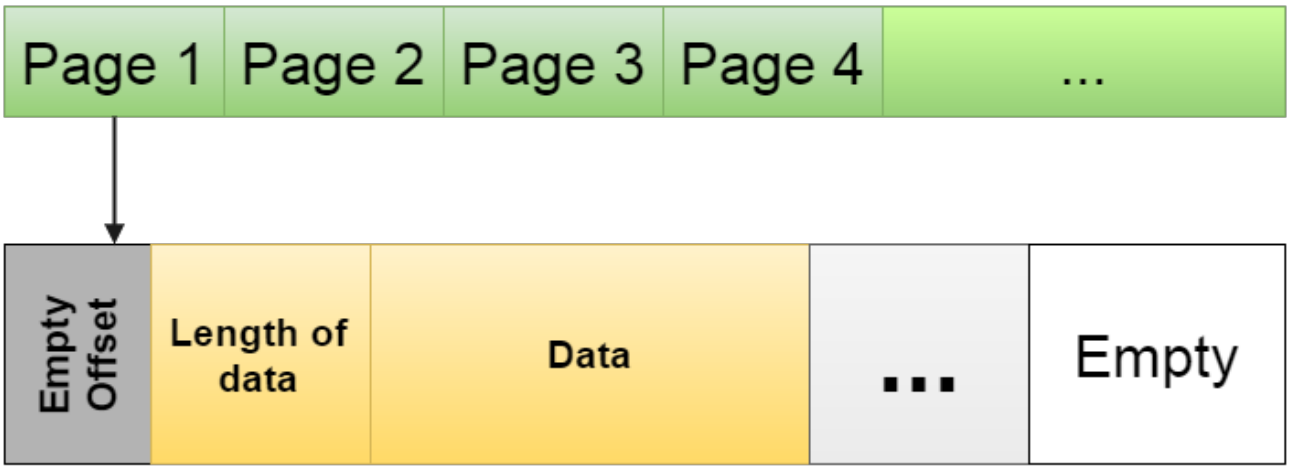

Figure 1: Disk Page Organisation

1.4 Class diagram

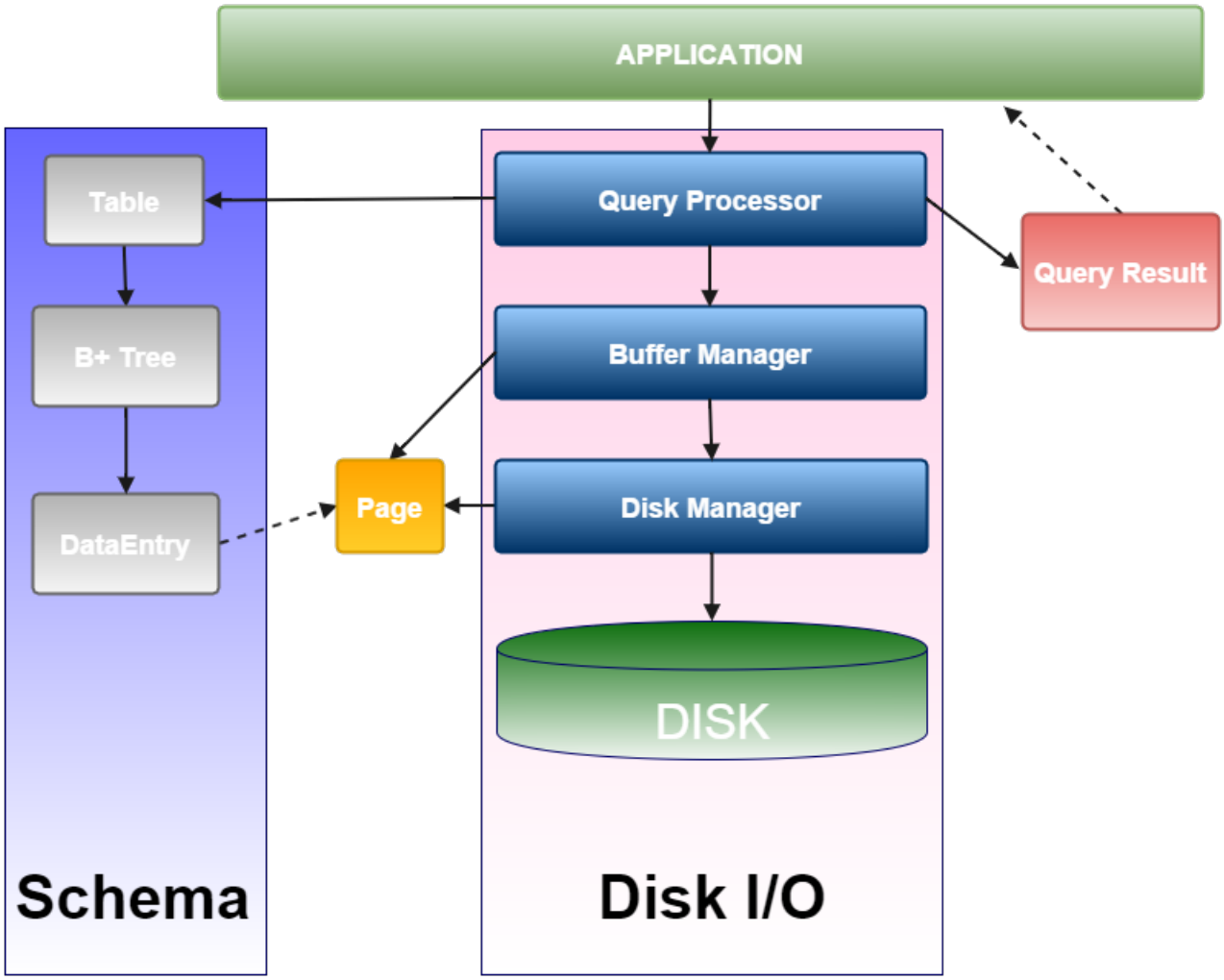

Figure 2: Class Diagram

References

[1] Concepts of Database Management. Philip J. Pratt, Joseph J. Adamski. Cengage Learning, 2008. ISBN: 1-4239-0147-9

[2] Database management systems. Raghu Ramakrishnan, Johannes Gehrke. McGraw-Hill., 2003. ISBN: 0-07-246563-8-ISBN 0-07-115110-9 (ISE)

[3] A First Course in Database Systems. Jeffrey D. Ullman, Jennifer Widom. Pearson Education., 2008. ISBN: 978-0-13-502176-7