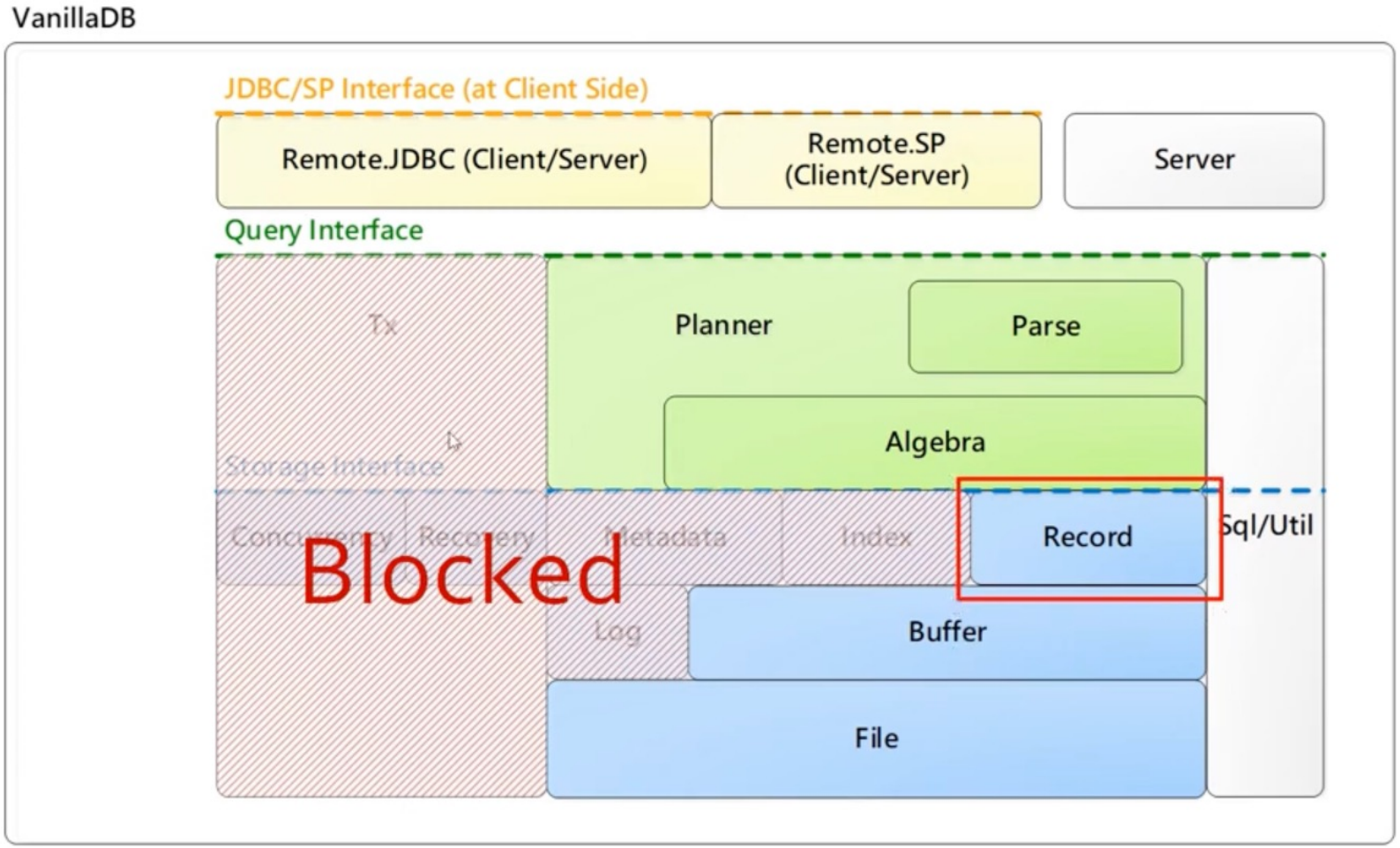
VanillaCore Walkthrough Part 5

Introduction to Databases

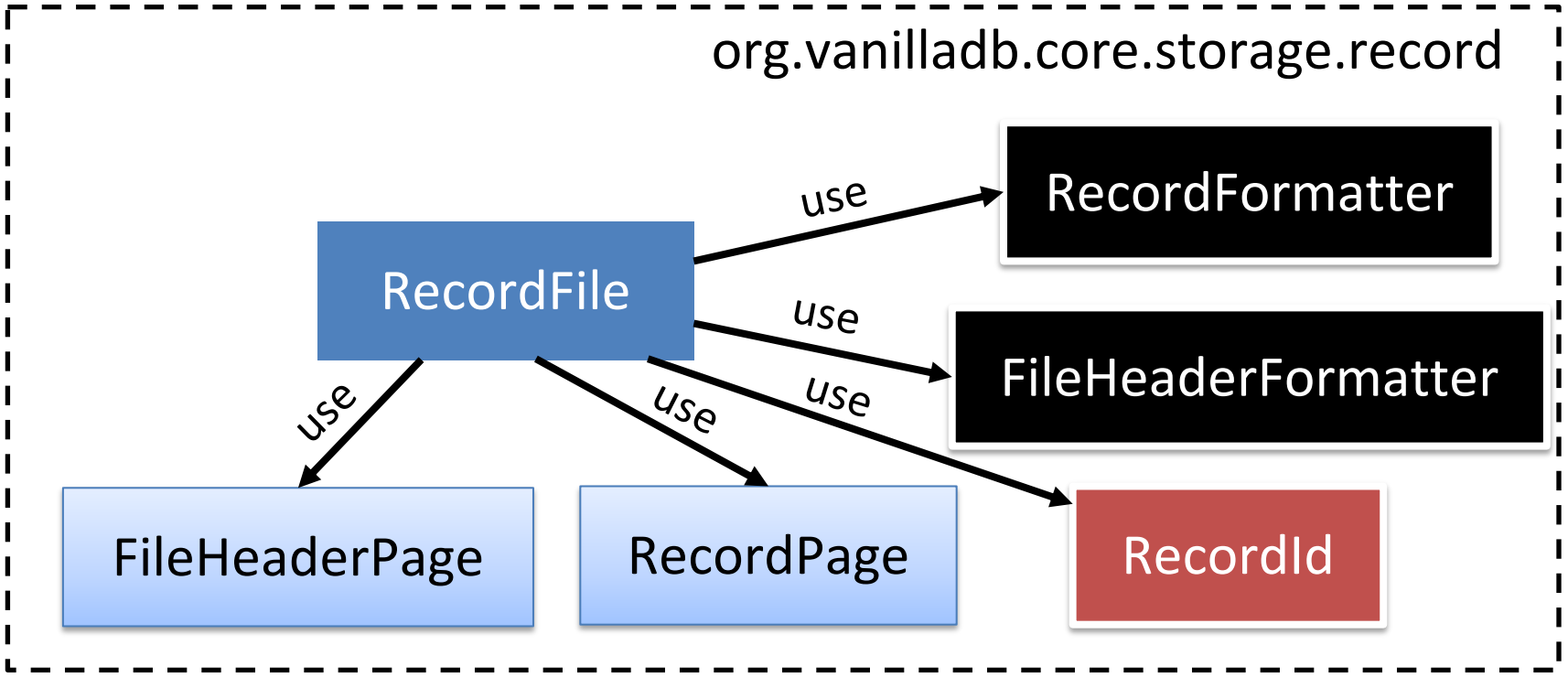
DataLab

CS, NTHU

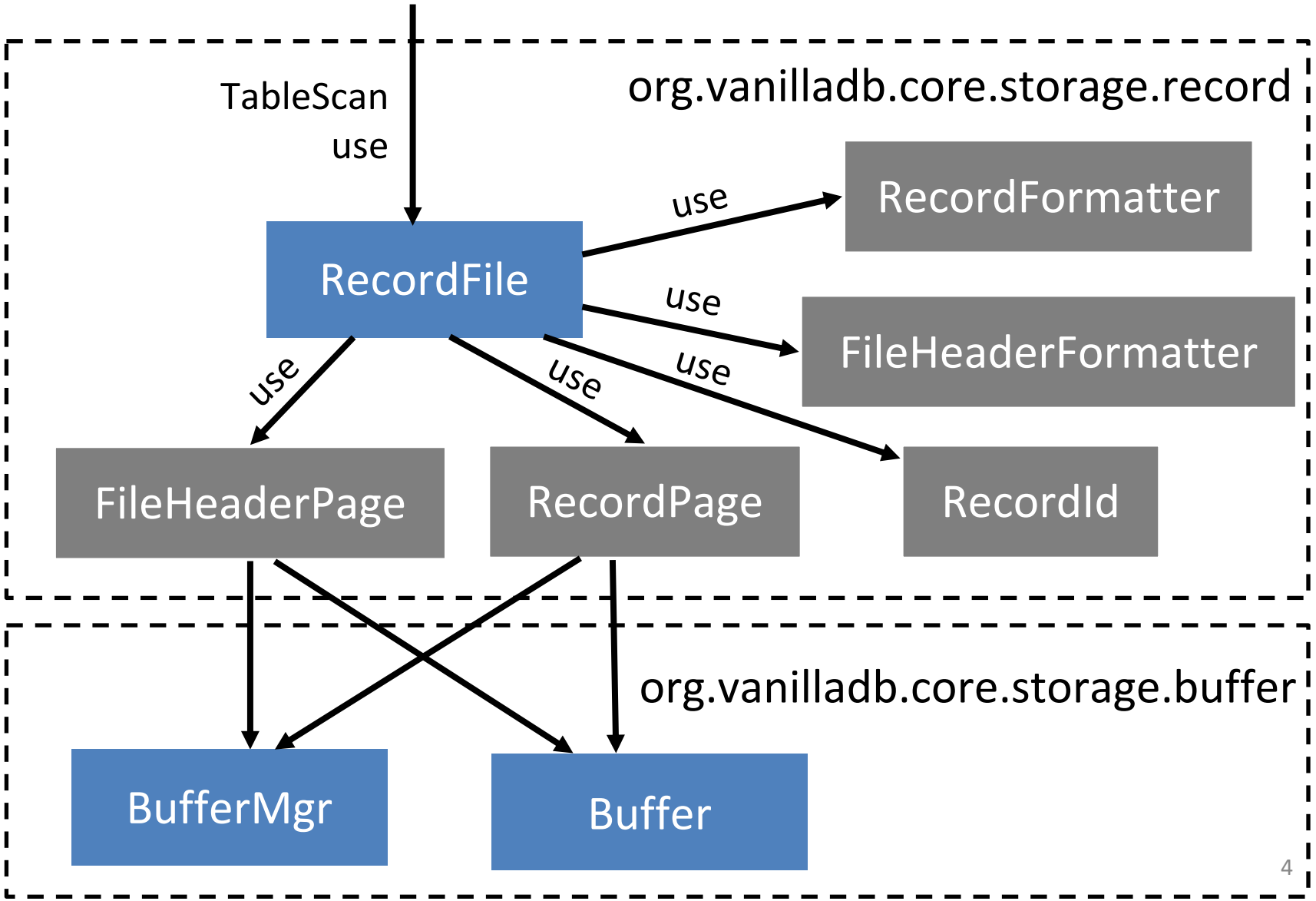
The Unlocked Modules



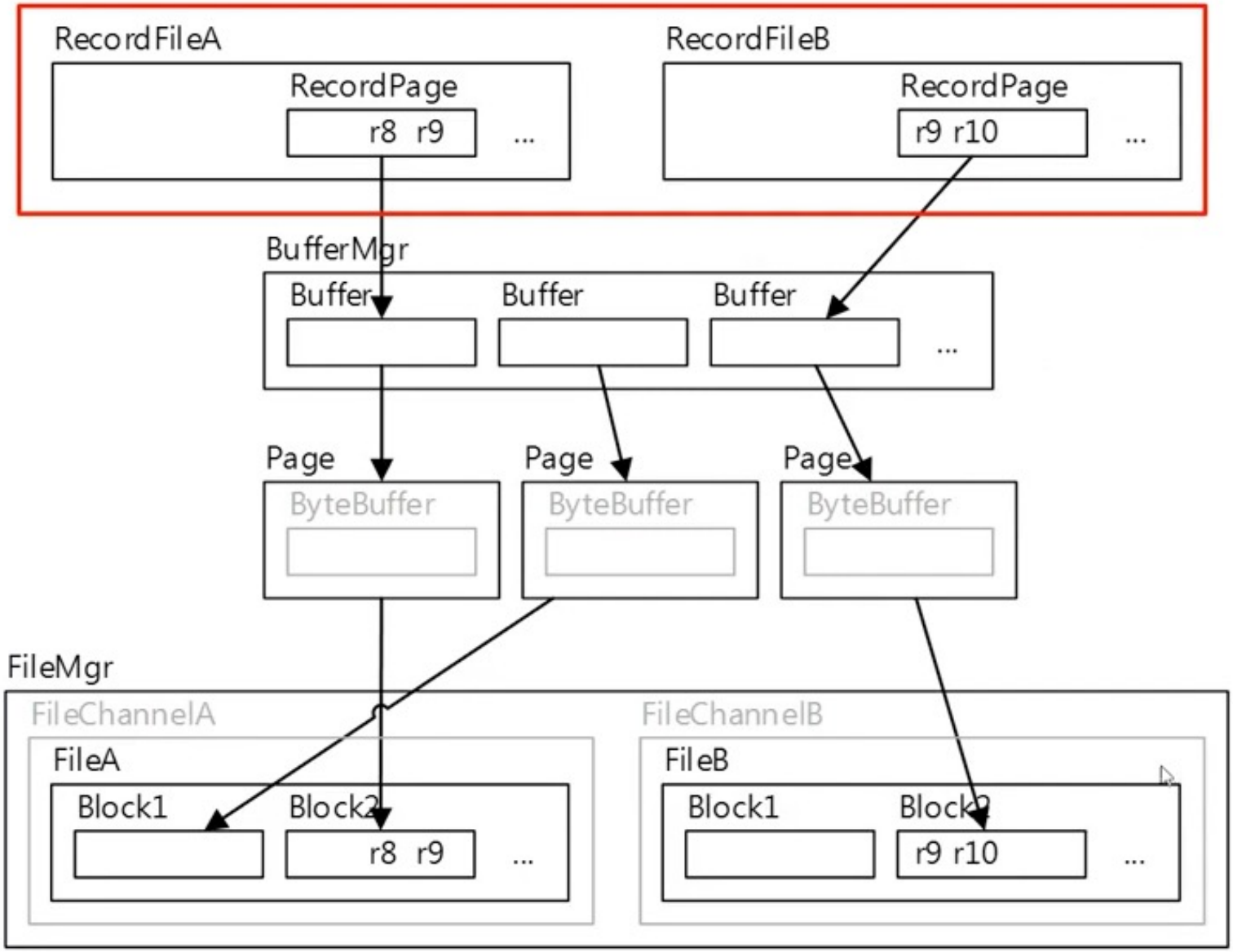
`record` Package



`record` Package

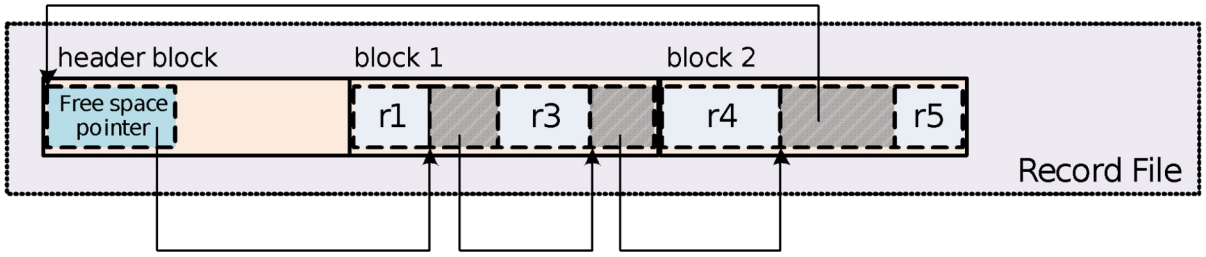


Recap of Data Access Layers

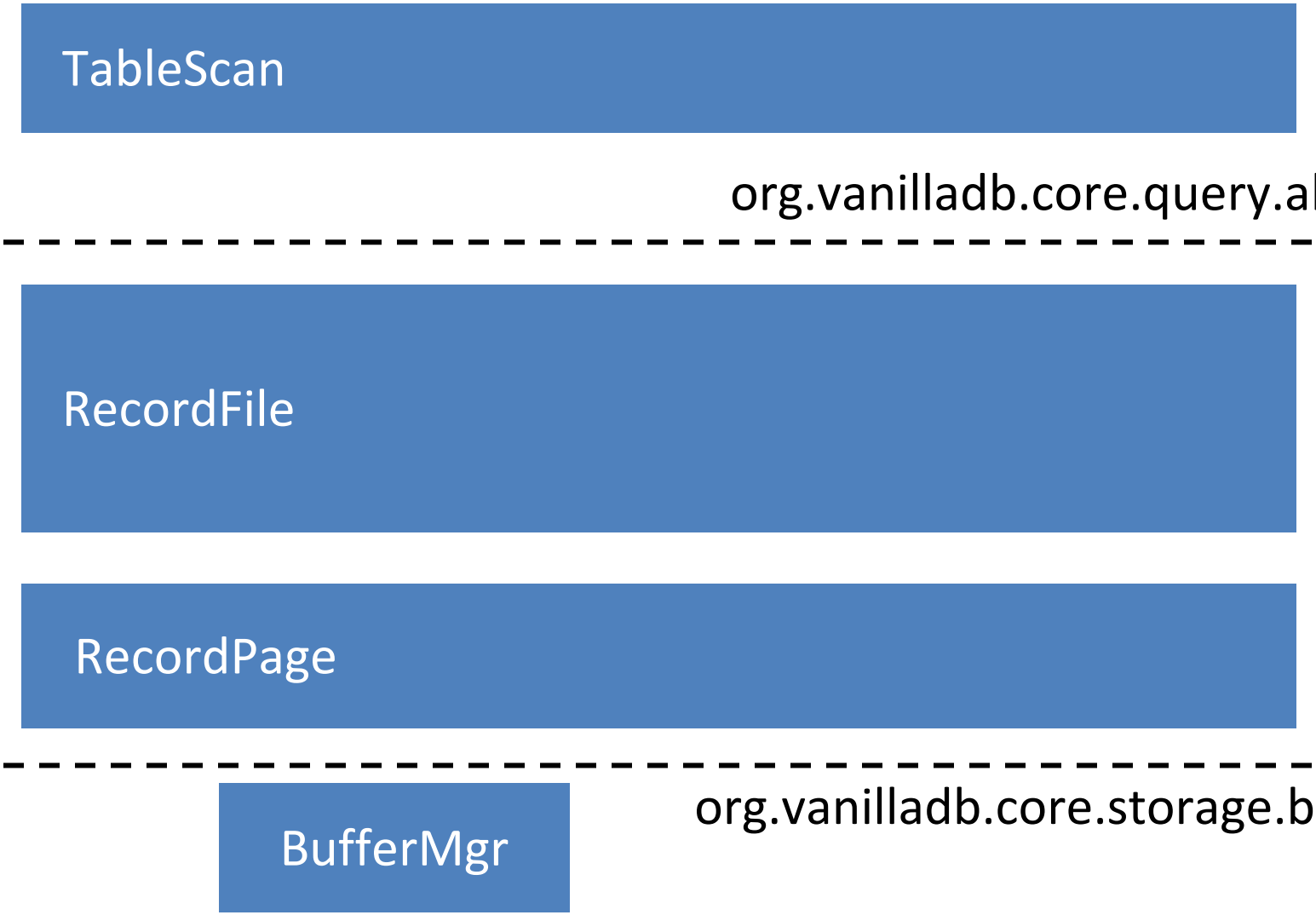


Responsibility of Each Component

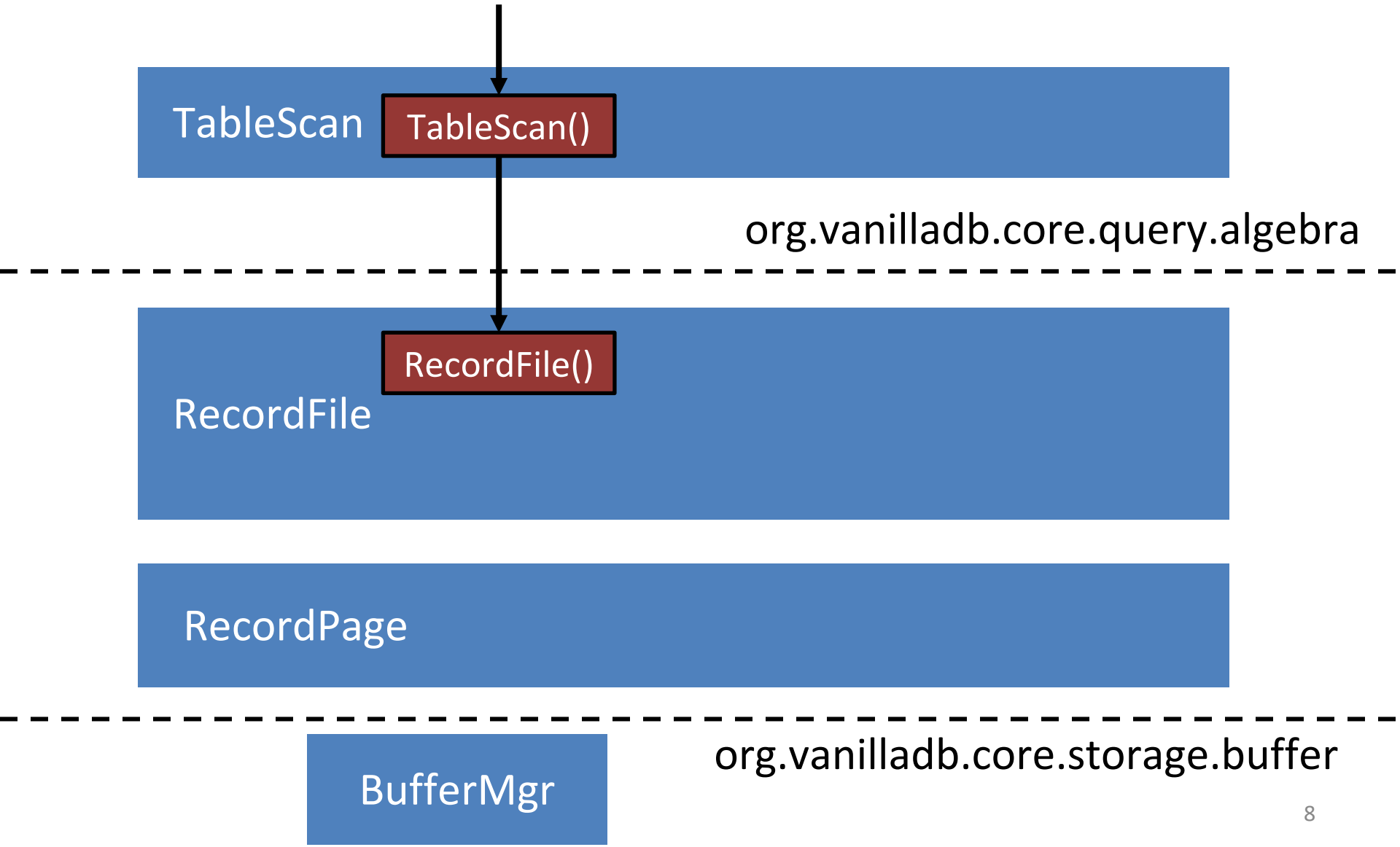
- `RecordFile`: manages a file of records and calls the concurrency manager to ensure isolation property
- `RecordPage`: lays out records in a page
- `FileHeaderPage`: header of free-space chain



How Is A Record Read?



How Is A Record Read?



TableScan

```
30  public class TableScan implements UpdateScan {
31      private RecordFile rf;
32      private Schema schema;
```

```
34      /**
35       * Creates a new table scan, and opens its corresponding record file.
36       *
37       * @param ti
38       *            the table's metadata
39       * @param tx
40       *            the calling transaction
41       */
42      public TableScan(TableInfo ti, Transaction tx) {
43          rf = ti.open(tx, true);
44          schema = ti.schema();
45      }
```

core-patch/src/main/java/org/vanilladb/core/query/algebra/TableScan.java

RecordFile

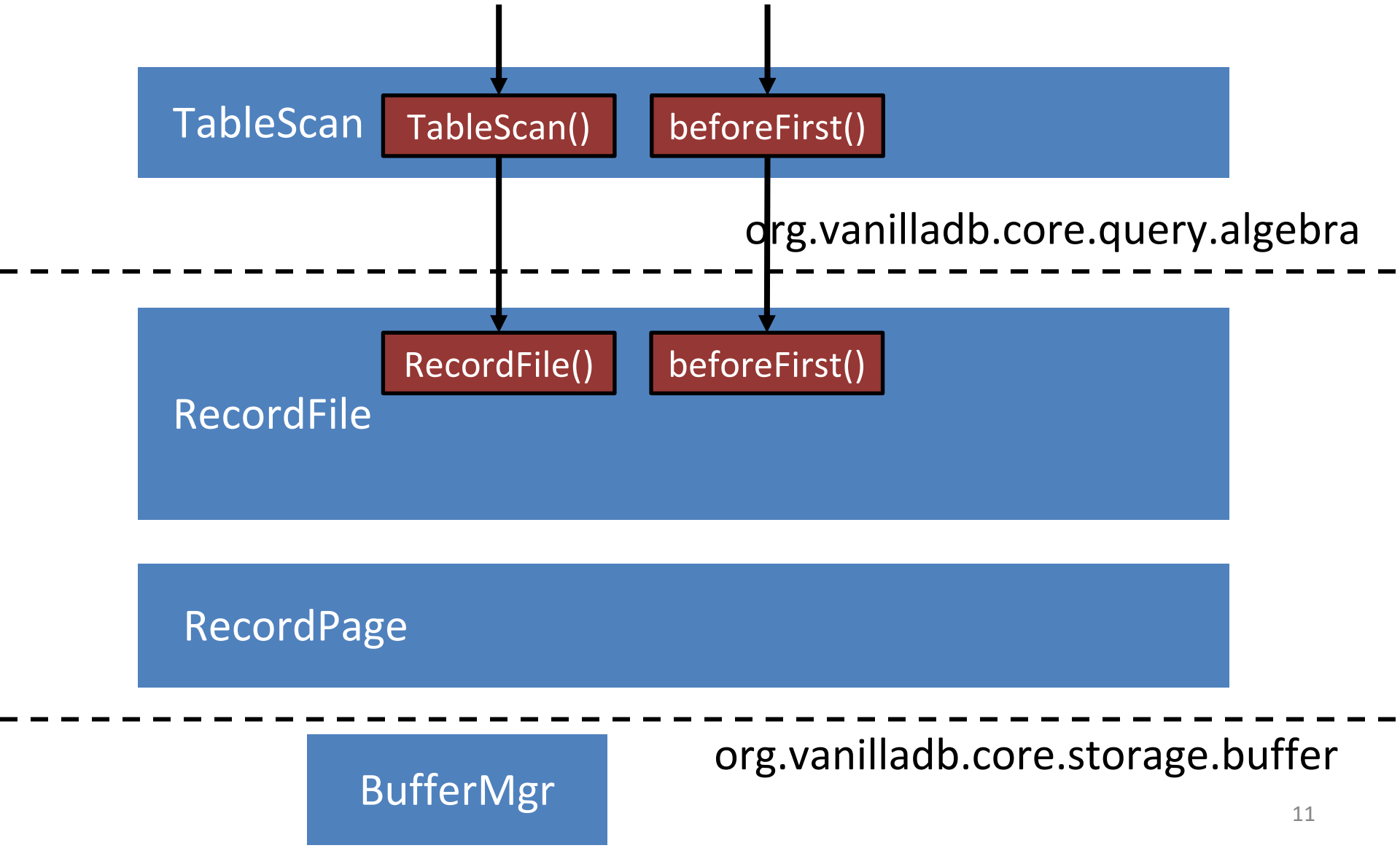
```java
public RecordFile open(Transaction tx, boolean doLog) {
    return new RecordFile(this, tx, doLog);
}
```

core-patch/src/main/java/org/vanilladb/core/storage/metadata/TableInfo.java

```java
68      public RecordFile(TableInfo ti, Transaction tx, boolean doLog) {
69          this.ti = ti;
70          this.tx = tx;
71          this.doLog = doLog;
72          fileName = ti.fileName();
73          headerBlk = new BlockId(fileName, blkNum:0);
74      }
```

core-patch/src/main/java/org/vanilladb/core/storage/record/RecordFile.java

How Is A Record Read?



beforeFirst()

(TableScan, RecordFile)

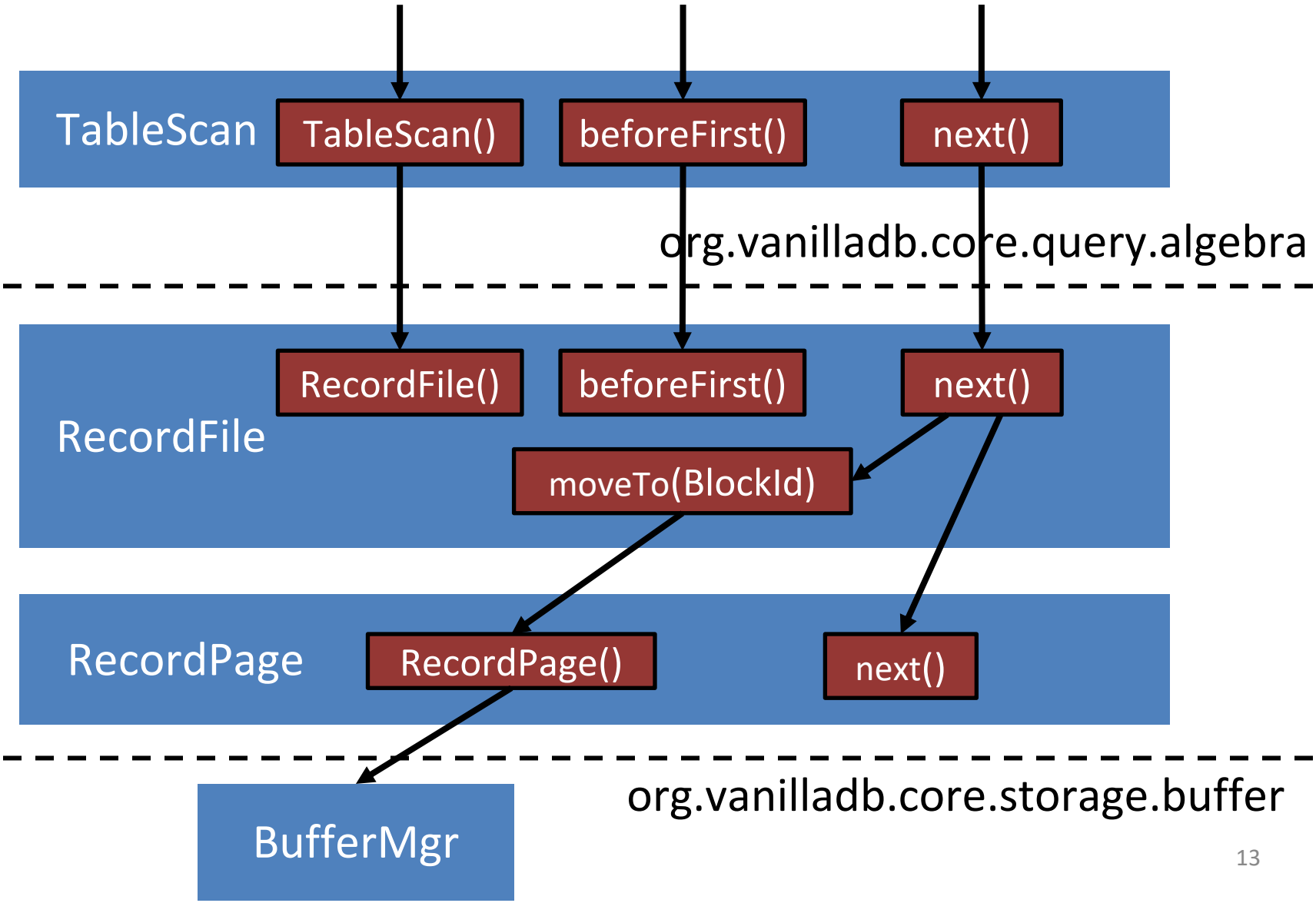
```
49      @Override
50      public void beforeFirst() {
51          rf.beforeFirst();
52      }
```

core-patch/src/main/java/org/vanilladb/core/query/algebra/TableScan.java

```
117     public void beforeFirst() {
118         close();
119         currentBlkNum = 0; // first data block is block 1
120         isBeforeFirsted = true;
121     }
```

core-patch/src/main/java/org/vanilladb/core/storage/record/RecordFile.java

How Is A Record Read?



next()

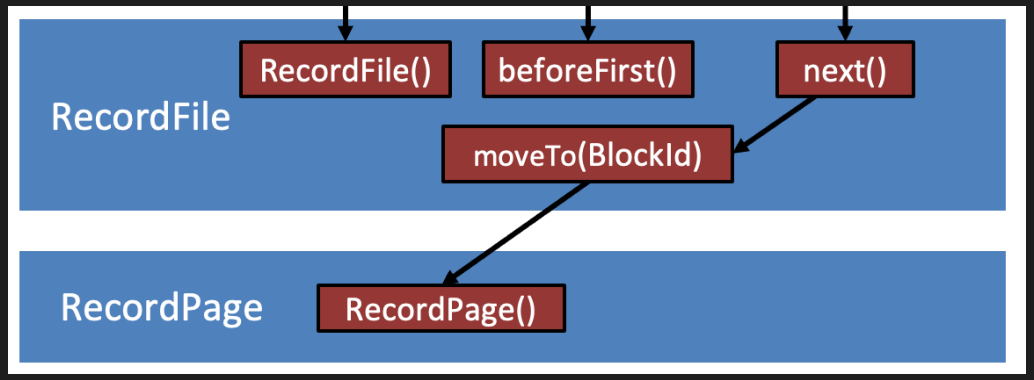
```
55      public boolean next() {
56          return rf.next();
57      }
```

core-patch/src/main/java/org/vanilladb/core/query/algebra/TableScan.java

```
128     public boolean next() {
129         if (!isBeforeFirsted)
130             throw new IllegalStateException("You must call beforeFirst() before iterating table '"
131                     + ti.tableName() + "'");
132
133         if (currentBlkNum == 0 && !moveTo(b:1))
134             return false;
135         while (true) {
136             if (rp.next())
137                 return true;
138             if (!moveTo(currentBlkNum + 1))
139                 return false;
140         }
141     }
```

core-patch/src/main/java/org/vanilladb/core/storage/record/RecordFile.java

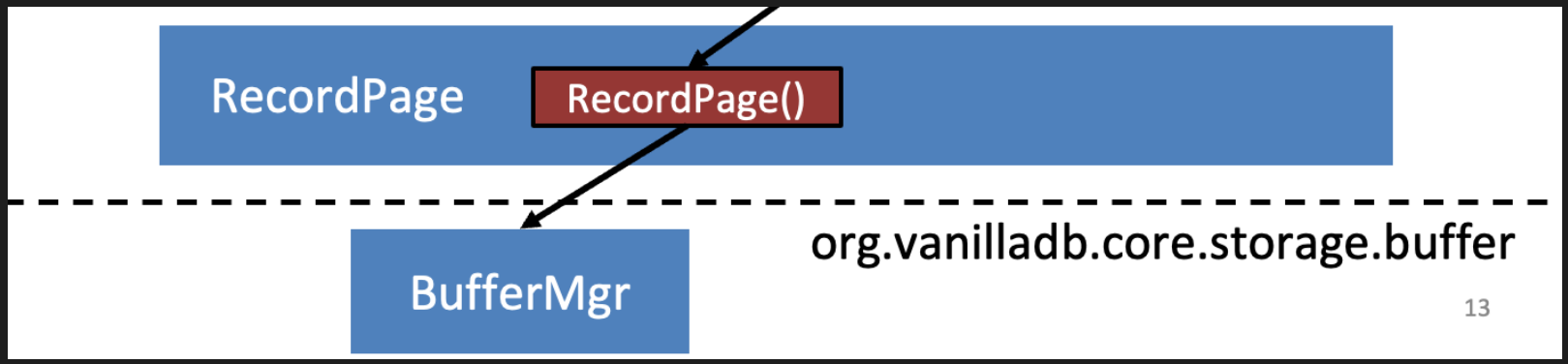
moveTo()




```
364     private boolean moveTo(long b) {
365         if (rp != null)
366             rp.close();
367
368         if (b >= fileSize()) // block b not allocated yet
369             return false;
370         currentBlkNum = b;
371         BlockId blk = new BlockId(fileName, currentBlkNum);
372         rp = new RecordPage(blk, ti, tx, doLog);
373         return true;
374     }
```

core-patch/src/main/java/org/vanilladb/core/storage/record/RecordFile.java

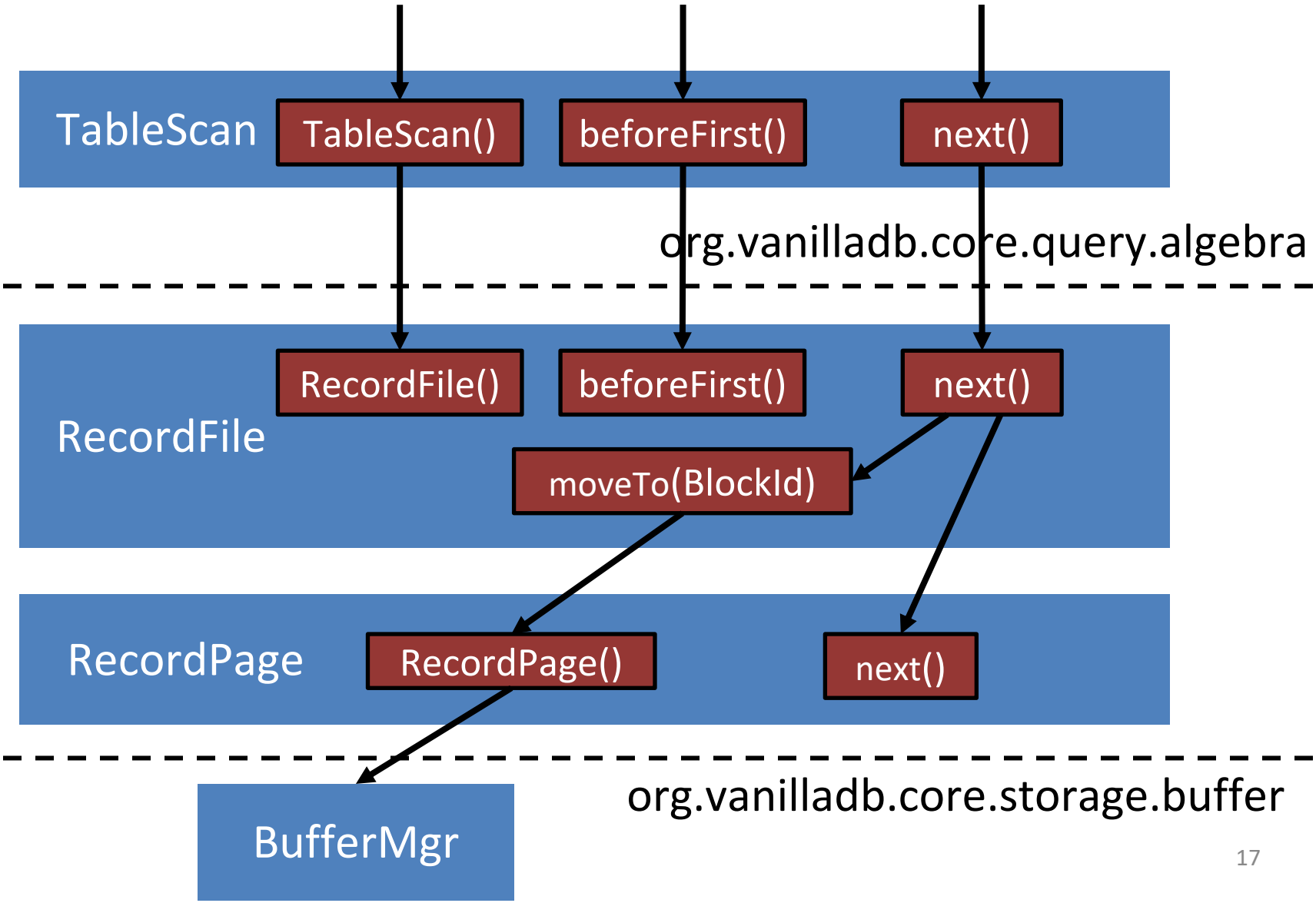
RecordPage




```
140     public RecordPage(BlockId blk, TableInfo ti, Transaction tx, boolean doLog) {
141         this.blk = blk;
142         this.tx = tx;
143         this.ti = ti;
144         this.doLog = doLog;
145         currentBuff = tx.bufferMgr().pin(blk);
```

core-patch/src/main/java/org/vanilladb/core/storage/record/RecordPage.java

How Is A Record Read?



next()
(RecordPage)

```
128     public boolean next() {
129         if (!isBeforeFirsted)
130             throw new IllegalStateException("You must call beforeFirst() before iterating table '"
131                     + ti.tableName() + "'");
132 
133         if (currentBlkNum == 0 && !moveTo(b:1))
134             return false;
135         while (true) {
136             if (rp.next())
137                 return true;
138             if (!moveTo(currentBlkNum + 1))
139                 return false;
140         }
141     }
```

core-patch/src/main/java/org/vanilladb/core/storage/record/RecordFile.java

next()
(RecordPage)

```
175     public boolean next() {
176         return searchFor(INUSE);
177     }
```

```
342     private boolean searchFor(int flag) {
343         currentSlot++;
344         while (isValidSlot()) {
345             if ((Integer) getVal(currentPos(), INTEGER).asJavaVal() == flag) {
346                 return true;
347             }
348             currentSlot++;
349         }
350         return false;
351     }
```

core-patch/src/main/java/org/vanilladb/core/storage/record/RecordPage.java

Review

```
VanillaDb.init("studentdb");

// Step 1 correspondence
Transaction tx = VanillaDb.txMgr().transaction(
Connection.TRANSACTION_SERIALIZABLE, true);

// Step 2 correspondence
Planner planner = VanillaDb.newPlanner();
String query = "SELECT s-name, d-name FROM departments, "
      + "students WHERE major-id = d-id";
Plan plan = planner.createQueryPlan(query, tx);
Scan scan = plan.open();

// Step 3 correspondence
System.out.println("name\tmajor");
System.out.println("-------\t-------");
while (scan.next()) {
      String sName = (String) scan.getVal("s-
      name").asJavaVal();
      String dName = (String) scan.getVal("d-
      name").asJavaVal();
      System.out.println(sName + "\t" + dName);
}
scan.close();

// Step 4 correspondence
tx.commit();
```

getVal()

```
69      @Override
70      public Constant getVal(String fldName) {
71          return rf.getVal(fldName);
72      }
```

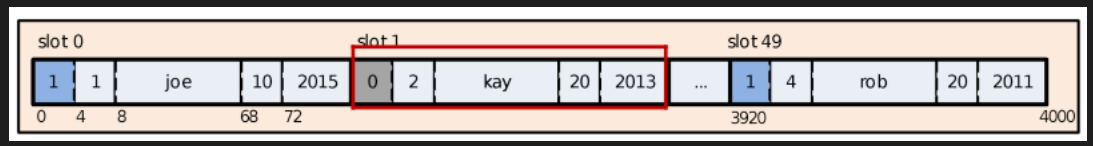
core-patch/src/main/java/org/vanilladb/core/query/algebra/TableScan.java

```
152     public Constant getVal(String fldName) {
153         return rp.getVal(fldName);
154     }
```

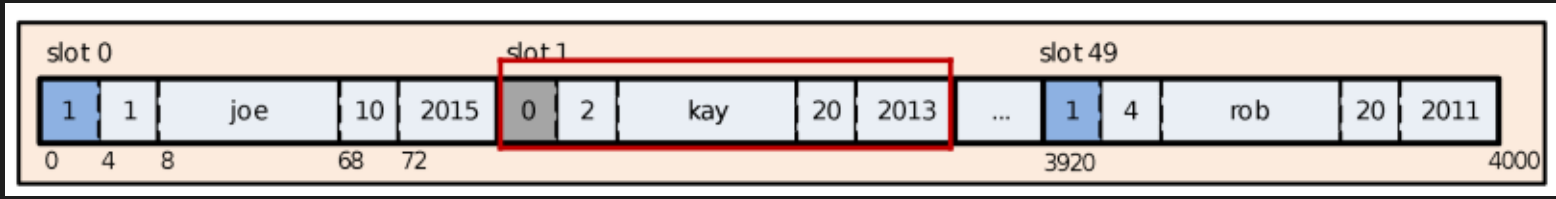
core-patch/src/main/java/org/vanilladb/core/storage/record/RecordFile.java

```
187     public Constant getVal(String fldName) {
188         int position = fieldPos(fldName);
189         return getVal(position, ti.schema().type(fldName));
190     }
```

core-patch/src/main/java/org/vanilladb/core/storage/record/RecordPage.java



fieldPos()

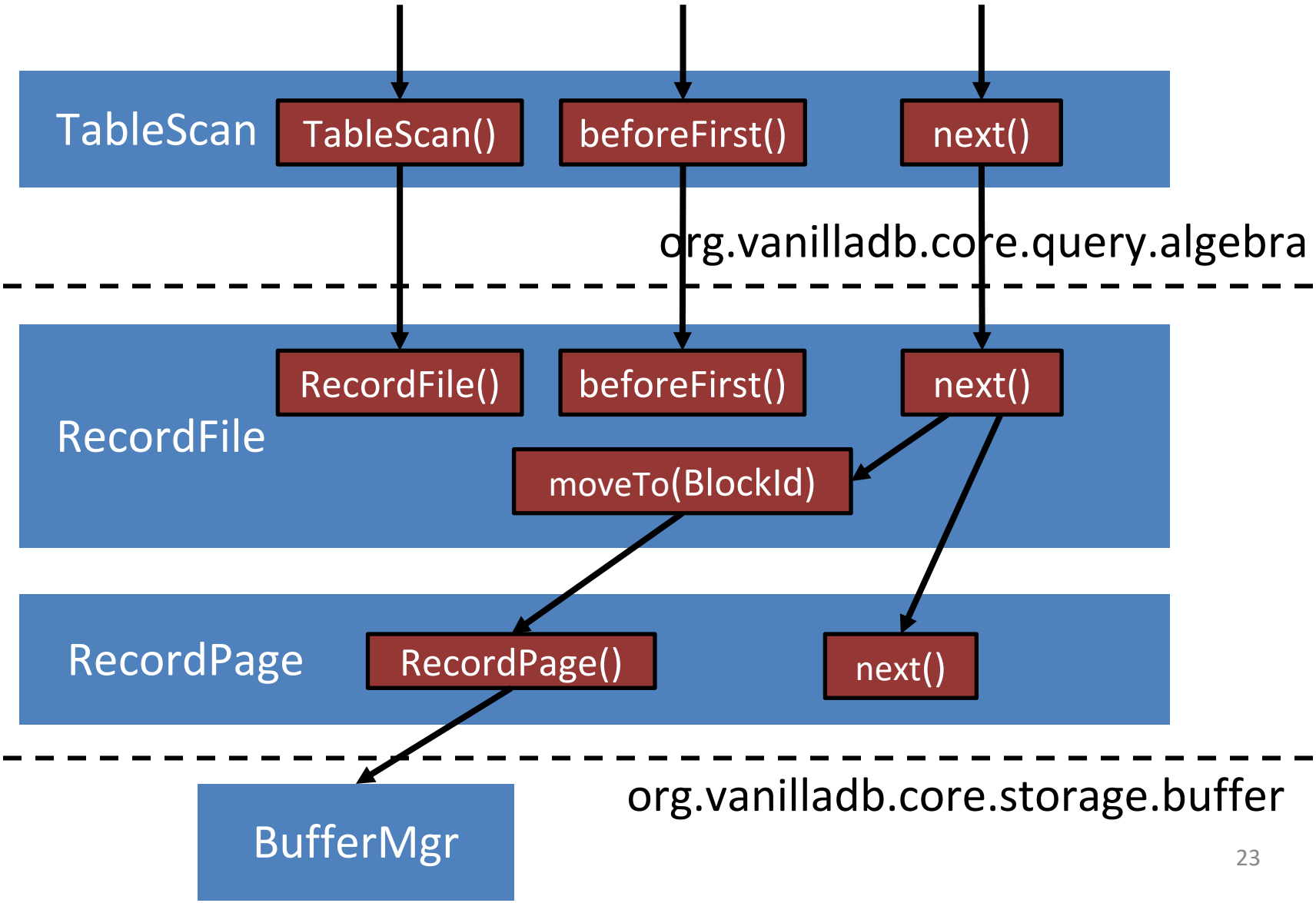



```
187     public Constant getVal(String fldName) {
188         int position = fieldPos(fldName);
189         return getVal(position, ti.schema().type(fldName));
190     }
```

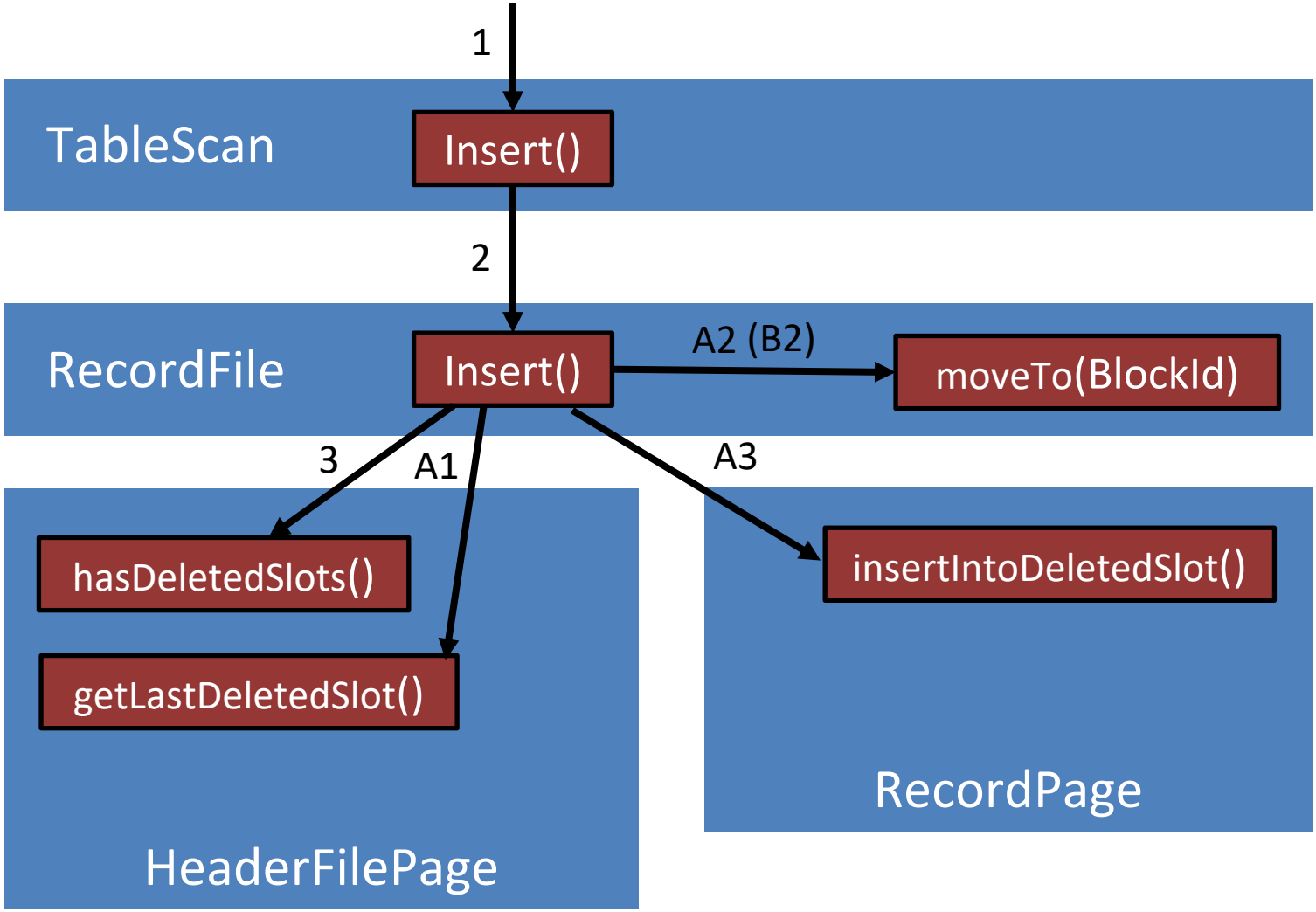
```
333     private int fieldPos(String fldName) {
334         int offset = FLAG_SIZE + myOffsetMap.get(fldName);
335         return currentPos() + offset;
336     }
```

core-patch/src/main/java/org/vanilladb/core/storage/record/RecordPage.java

How Is A Record Read?



How Is A Record Inserted?



Insert()

```
100     @Override
101     public void insert() {
102         rf.insert();
103     }
```

core-patch/src/main/java/org/vanilladb/core/query/algebra/TableScan.java

```
221     public void insert() {
222         // Block read-only transaction
223         if (tx.isReadOnly() && !isTempTable())
224             throw new UnsupportedOperationException();
225
226         // Insertion may change the properties of this file,
227         // so that we need to lock the file.
228         if (!isTempTable())
229             tx.concurrencyMgr().modifyFile(fileName);
230
231         // Modify the free chain which is start from a pointer in
232         // the header of the file.
233         if (fhp == null)
234             fhp = openHeaderForModification();
```

core-patch/src/main/java/org/vanilladb/core/storage/record/RecordFile.java

Insert()

```
239         if (fhp.hasDeletedSlots()) {
240             // Insert into a deleted slot
241             moveToRecordId(fhp.getLastDeletedSlot());
242             RecordId lds = rp.insertIntoDeletedSlot();
243             fhp.setLastDeletedSlot(lds);
```

core-patch/src/main/java/org/vanilladb/core/storage/record/RecordFile.java

```
106     public RecordId getLastDeletedSlot() {
107         Constant blkNum = getVal(OFFSET_LDS_BLOCKID, BIGINT);
108         Constant rid = getVal(OFFSET_LDS_RID, INTEGER);
109         BlockId bid = new BlockId(fileName, (Long) blkNum.asJavaVal());
110         return new RecordId(bid, (Integer) rid.asJavaVal());
111     }
```

core-patch/src/main/java/org/vanilladb/core/storage/record/FileHeaderPage.java

```
24  public class RecordId implements Comparable<RecordId> {
25      private BlockId blk;
26      private int id;
```

core-patch/src/main/java/org/vanilladb/core/storage/record/RecordId.java

Insert()

```
239         if (fhp.hasDeletedSlots()) {
240             // Insert into a deleted slot
241             moveToRecordId(fhp.getLastDeletedSlot());
242             RecordId lds = rp.insertIntoDeletedSlot();
243             fhp.setLastDeletedSlot(lds);
```

core-patch/src/main/java/org/vanilladb/core/storage/record/RecordFile.java

```
336     public void moveToRecordId(RecordId rid) {
337         moveTo(rid.block().number());
338         rp.moveToId(rid.id());
339     }
```

core-patch/src/main/java/org/vanilladb/core/storage/record/RecordFile.java

```
272     public void moveToId(int id) {
273         currentSlot = id;
274     }
```

core-patch/src/main/java/org/vanilladb/core/storage/record/RecordPage.java

Insert()

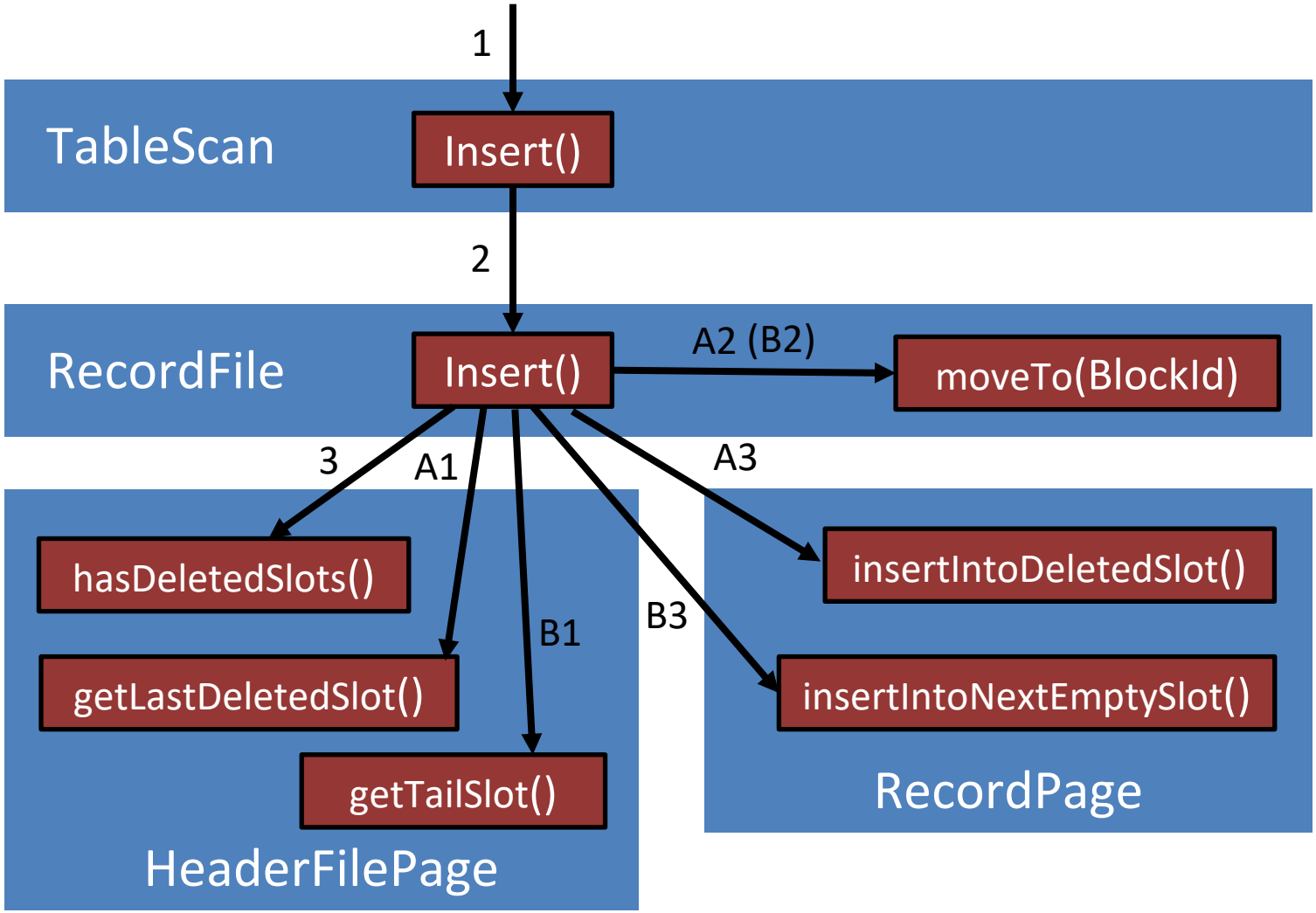
```
239             if (fhp.hasDeletedSlots()) {
240                 // Insert into a deleted slot
241                 moveToRecordId(fhp.getLastDeletedSlot());
242                 RecordId lds = rp.insertIntoDeletedSlot();
243                 fhp.setLastDeletedSlot(lds);
```

core-patch/src/main/java/org/vanilladb/core/storage/record/RecordFile.java

```
254     public RecordId insertIntoDeletedSlot() {
255         RecordId nds = getNextDeletedSlotId();
256         // Important: Erase the free chain information.
257         // If we didn't do this, it would crash when
258         // a tx try to set a VARCHAR at this position
259         // since the getVal would get negative size.
260         setNextDeletedSlotId(new RecordId(new BlockId(fileName:"", blkNum:0), id:0));
261         Constant flag = INUSE_CONST;
262         setVal(currentPos(), flag);
263         return nds;
264     }
```

core-patch/src/main/java/org/vanilladb/core/storage/record/RecordPage.java

How Is A Record Inserted?



Insert()

```
244         } else {
245             // Insert into a empty slot
246             if (!fhp.hasDataRecords()) { // no record inserted before
247                 // Create the first data block
248                 appendBlock();
249                 moveTo(b:1);
250                 rp.insertIntoNextEmptySlot();
251             } else {
252                 // Find the tail page
253                 RecordId tailSlot = fhp.getTailSolt();
254                 moveToRecordId(tailSlot);
255                 while (!rp.insertIntoNextEmptySlot()) {
256                     if (atLastBlock())
257                         appendBlock();
258                     moveTo(currentBlkNum + 1);
259                 }
260             }
261             fhp.setTailSolt(currentRecordId());
262         }
```

core-patch/src/main/java/org/vanilladb/core/storage/record/RecordFile.java

setVal()

```
90      @Override
91      public void setVal(String fldName, Constant val) {
92          rf.setVal(fldName, val);
93      }
```

core-patch/src/main/java/org/vanilladb/core/query/algebra/TableScan.java

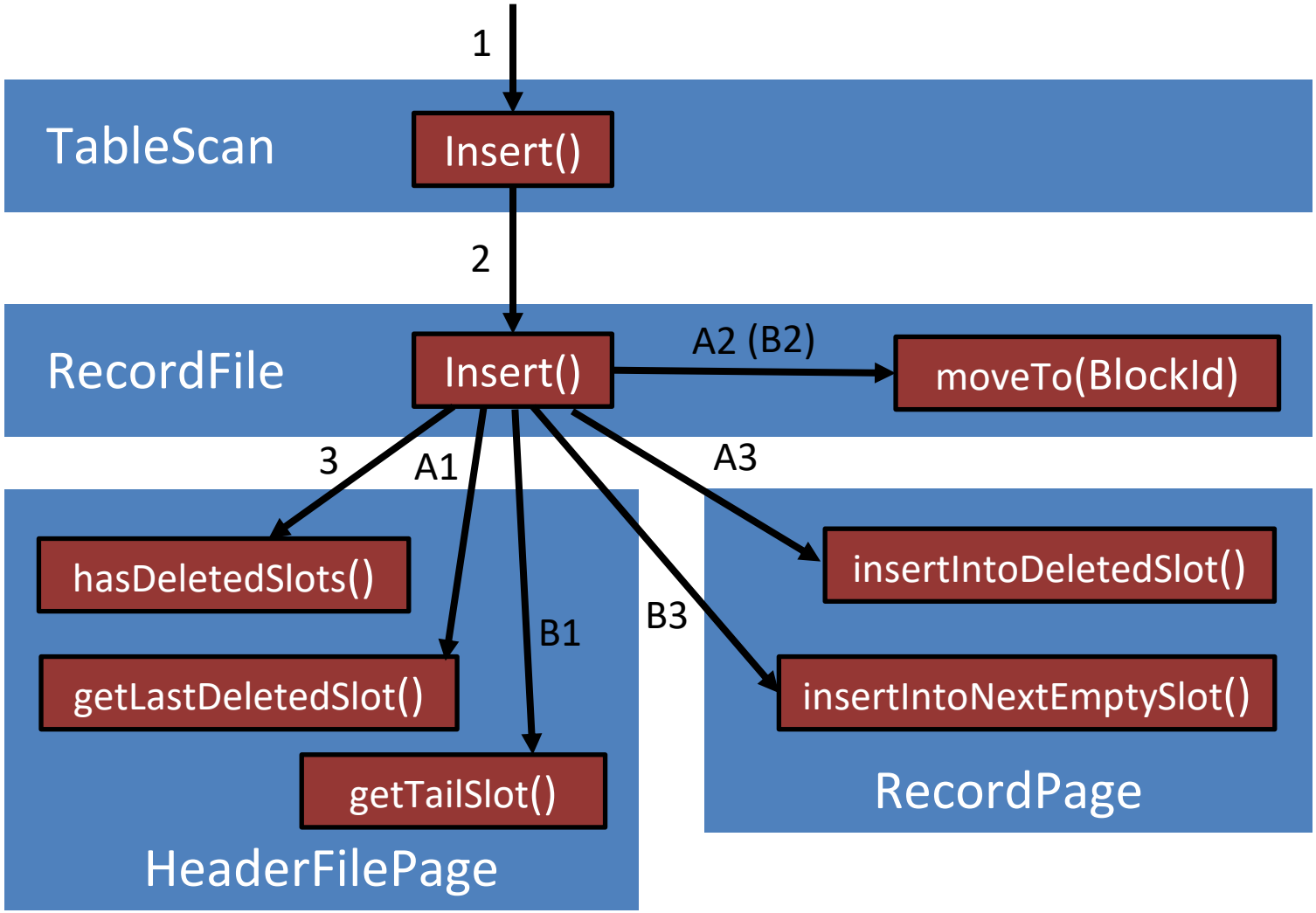
```
165     public void setVal(String fldName, Constant val) {
166         if (tx.isReadOnly() && !isTempTable())
167             throw new UnsupportedOperationException();
168         Type fldType = ti.schema().type(fldName);
169
170         Constant v = val.castTo(fldType);
171         if (Page.size(v) > Page.maxSize(fldType))
172             throw new SchemaIncompatibleException();
173         rp.setVal(fldName, v);
174     }
```

core-patch/src/main/java/org/vanilladb/core/storage/record/RecordFile.java

```
200     public void setVal(String fldName, Constant val) {
201         int position = fieldPos(fldName);
202         setVal(position, val);
203     }
```

core-patch/src/main/java/org/vanilladb/core/storage/record/RecordPage.java

How Is A Record Inserted?



Assigned Reading

- How is a record deleted?
- How is the free space chain maintained?