Retrieving Big Data

SDSC SAN DIEGO SUPERCOMPUTER CENTER

On Counting and Distinct

- **Count the number of Drinkers**

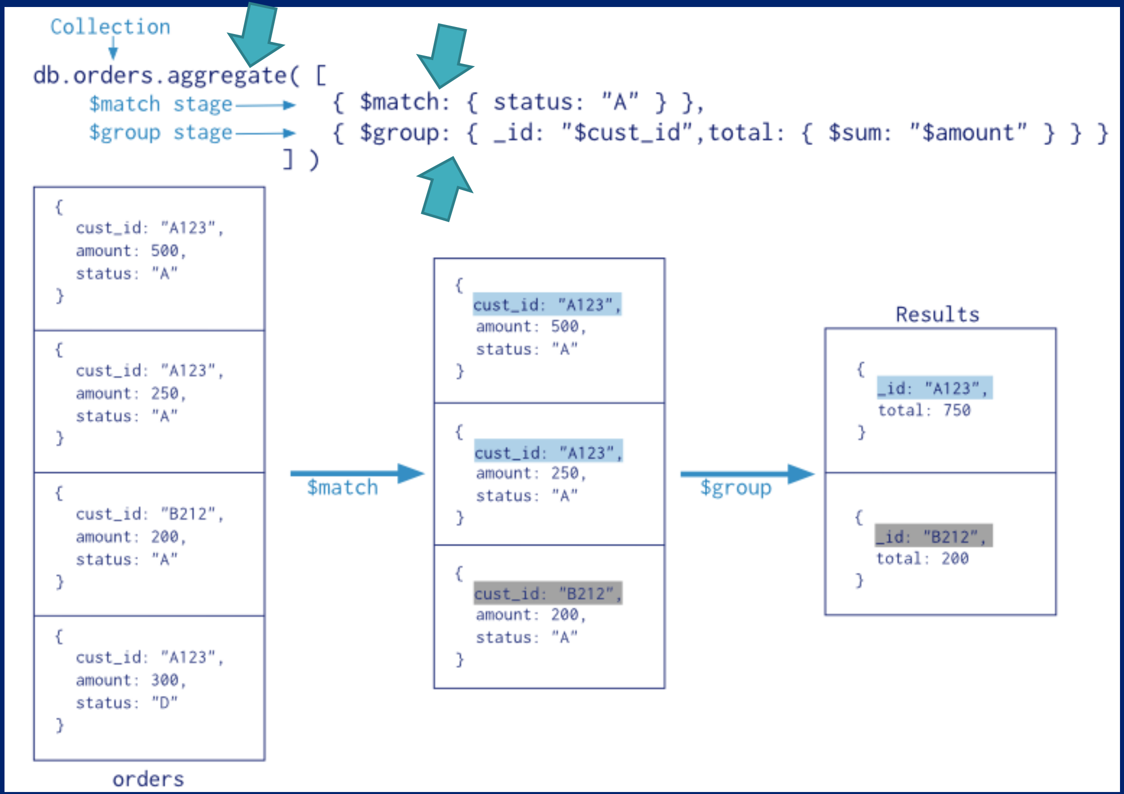
  Select count(*) from Drinkers

  - db.Drinkers.count()
- **Count the number of unique addresses of Drinkers**

  Select count(distinct addr) from Drinkers

  - db.Drinkers.count(addr: {$exists: true})
- **Get the distinct values of an array**
  - Data: {_id: 1, places: [USA, France, USA, Spain, UK, Spain]
  - db.countryDB.distinct(places)
    - [USA, France, Spain, UK]
  - db.countryDB.distinct(places).length
    - 4

Aggregation Framework



```
db.orders.aggregate( [
    { $match: { status: "A" } },
    { $group: { _id: "$cust_id",total: { $sum: "$amount" } } }
] )
```

```
{
  cust_id: "A123",
  amount: 500,
  status: "A"
}
{
  cust_id: "A123",
  amount: 250,
  status: "A"
}
{
  cust_id: "B212",
  amount: 200,
  status: "A"
}
{
  cust_id: "A123",
  amount: 300,
  status: "D"
}
```

orders

```
{
  cust_id: "A123",
  amount: 500,
  status: "A"
}
{
  cust_id: "A123",
  amount: 250,
  status: "A"
}
{
  cust_id: "B212",
  amount: 200,
  status: "A"
}
```

Results

```
{
  _id: "A123",
  total: 750
}
{
  _id: "B212",
  total: 200
}
```

- Role of aggregation framework
  - Grouping, aggregate functions, sorting, …

Multi-attribute Grouping

```
db.computers.aggregate(
    [
        {
            $group : {
                _id : { brand: "$brand", title: "$title", category: "$category", code: "$code" },
                count: { $sum: 1 }
            }
        }
        {
            $sort: { count:  1, category: -1}
        }
    ]
)
```

Text Search with Aggregation

```
db.articles.aggregate(
  [
    { $match: { $text: { $search: “Hillary Democrat" } } },
    { $sort: { score: { $meta: "textScore" } } },
    { $project: { title: 1, _id: 0 } }
  ]
)
```

Join in MongoDB

orders

```
{ "_id" : 1, "item" : "abc", "price" : 12, "quantity" : 2 }
{ "_id" : 2, "item" : "jkl", "price" : 20, "quantity" : 1 }
{ "_id" : 3  }
```

```
db.orders.aggregate([
  {    $lookup:      {
         from: "inventory",
         localField: "item",
         foreignField: "sku",
         as: "inventory_docs"
       }
  }
])
```

inventory

```
{ "_id" : 1, "sku" : "abc", description: "product 1", "instock" : 120 }
{ "_id" : 2, "sku" : "def", description: "product 2", "instock" : 80 }
{ "_id" : 3, "sku" : "ijk", description: "product 3", "instock" : 60 }
{ "_id" : 4, "sku" : "jkl", description: "product 4", "instock" : 70 }
{ "_id" : 5, "sku": null, description: "Incomplete" }
{ "_id" : 6 }
```

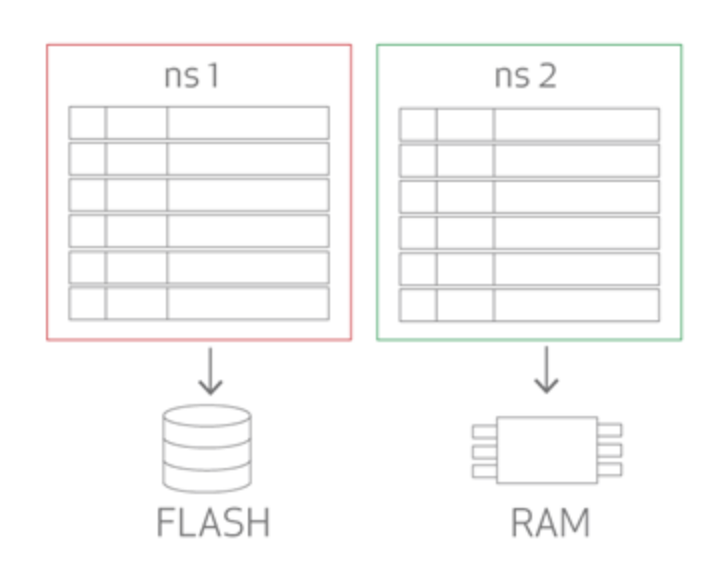
```
{
  "_id" : 1,
   "item" : "abc",
  "price" : 12,
  "quantity" : 2,
  "inventory_docs" : [{ "_id" : 1, "sku" : "abc", description: "product 1", "instock" : 120 }]
}
{
  "_id" : 2,
  "item" : "jkl",
  "price" : 20,
  "quantity" : 1,
  "inventory_docs" : [{ "_id" : 4, "sku" : "jkl", "description" : "product 4", "instock" : 70 }]
}
{
  "_id" : 3,
  "inventory_docs" : [{ "_id" : 5, "sku" : null, "description" : "Incomplete" }, { "_id" : 6 }]
}
```

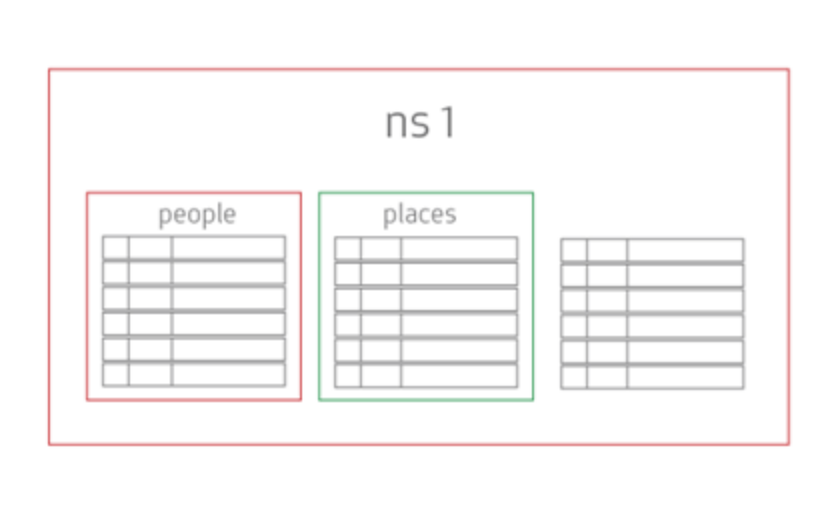
```
{ "_id" : 5, "sku": null, description: "Incomplete" }
{ "_id" : 6 }
```

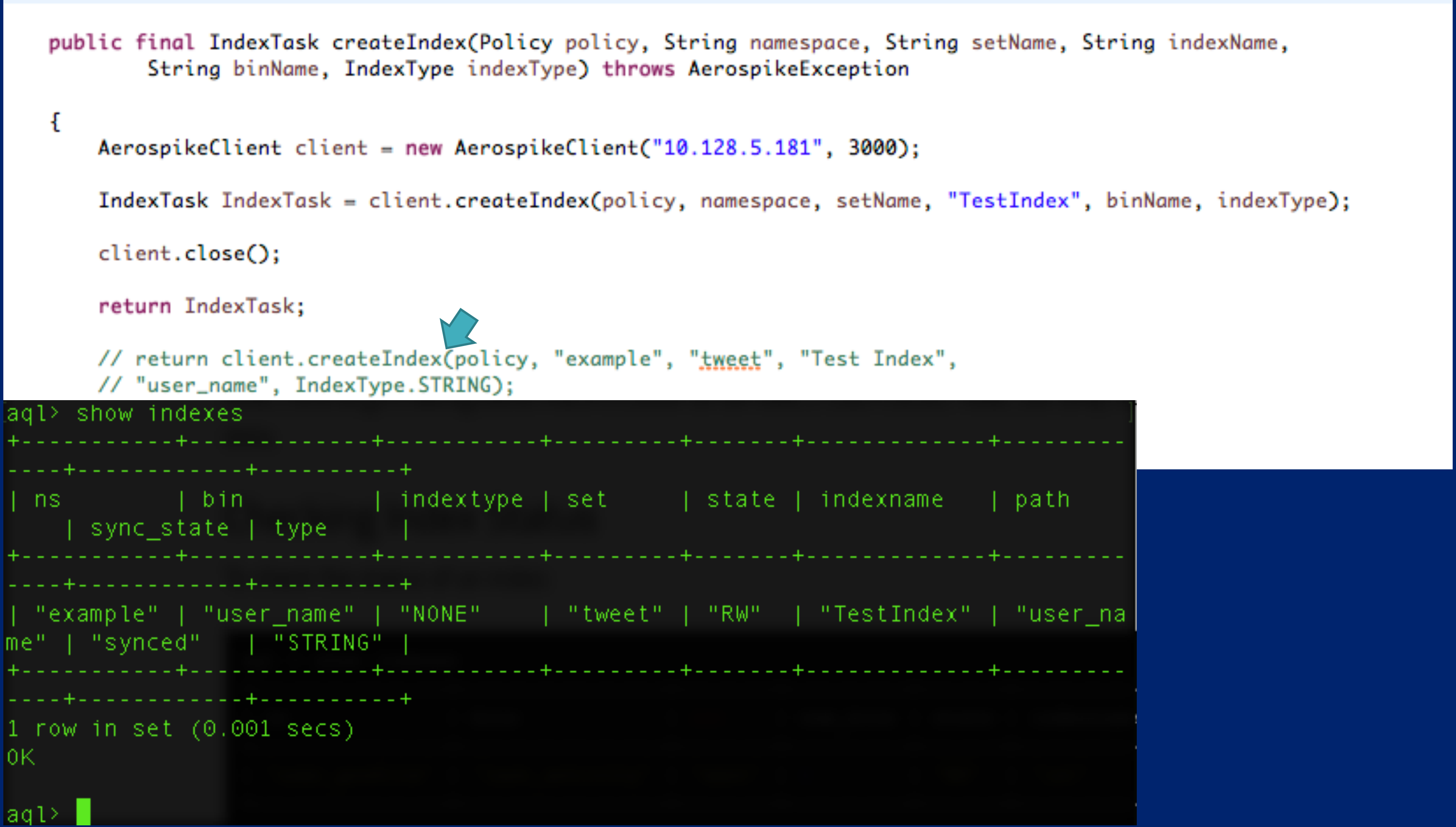
```
{ "_id" : 3  }
```

Pause

Querying Aerospike

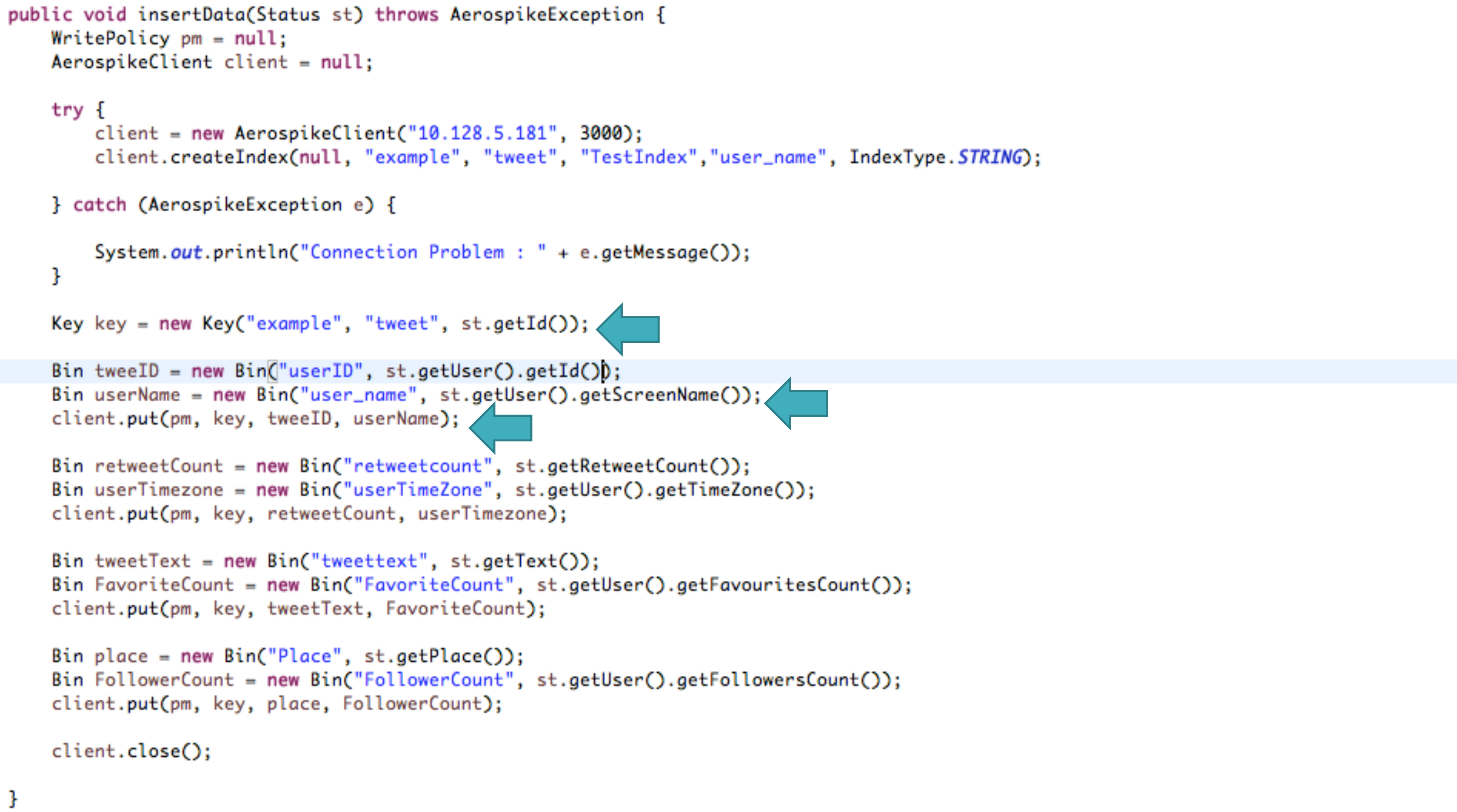




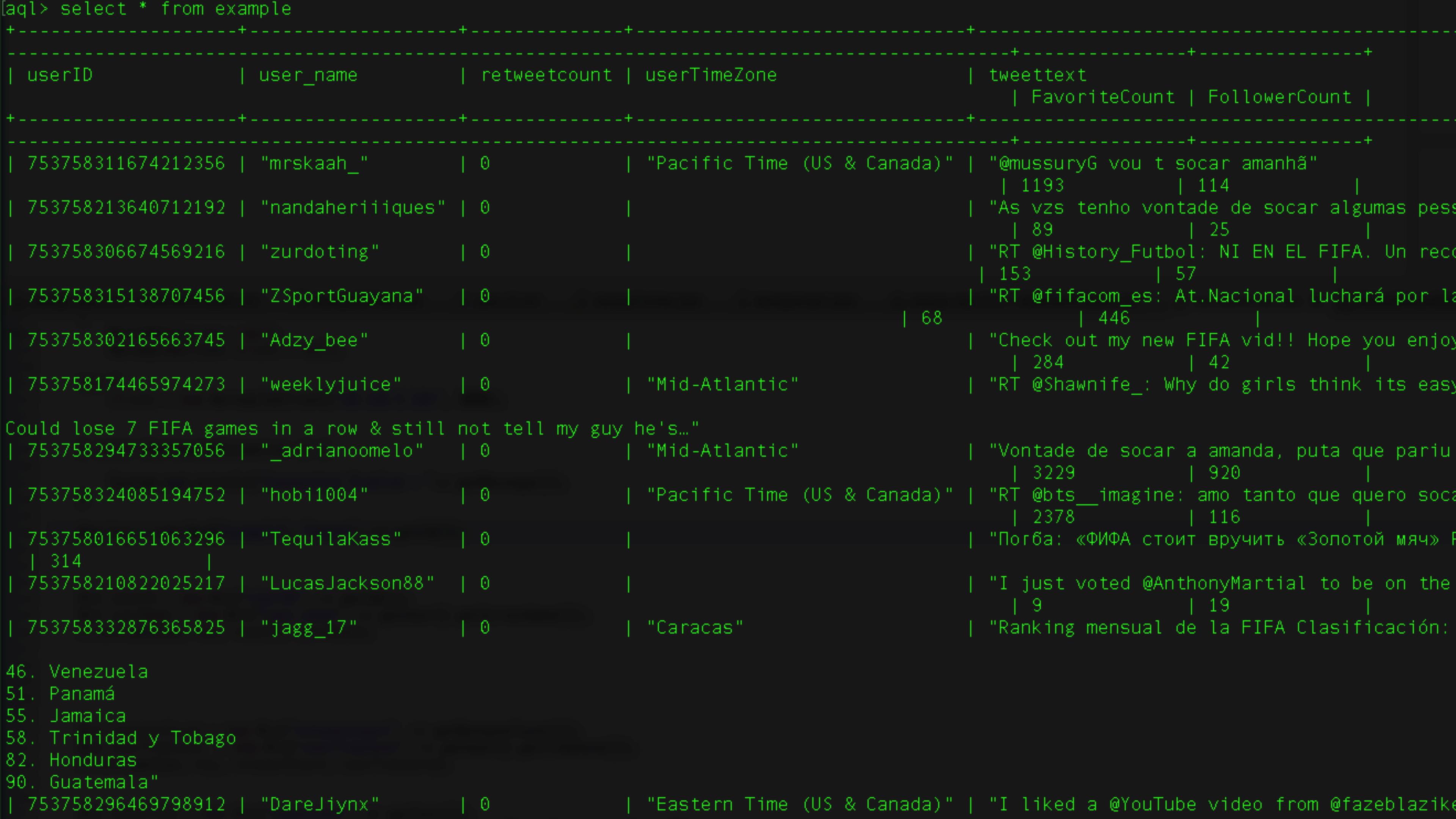


```
public final IndexTask createIndex(Policy policy, String namespace, String setName, String indexName,
        String binName, IndexType indexType) throws AerospikeException

{
    AerospikeClient client = new AerospikeClient("10.128.5.181", 3000);

    IndexTask IndexTask = client.createIndex(policy, namespace, setName, "TestIndex", binName, indexType);

    client.close();

    return IndexTask;

    // return client.createIndex(policy, "example", "tweet", "Test Index",
    // "user_name", IndexType.STRING);
```

```
aql> show indexes
+------------+--------------+------------+----------+-------+-------------+---------
----+------------+----------+
| ns         | bin          | indextype | set      | state | indexname   | path
    | sync_state | type     |
+------------+--------------+------------+----------+-------+-------------+---------
----+------------+----------+
| "example" | "user_name" | "NONE"     | "tweet" | "RW"  | "TestIndex" | "user_na
me" | "synced"   | "STRING" |
+------------+--------------+------------+----------+-------+-------------+---------
----+------------+----------+
1 row in set (0.001 secs)
OK

aql>
```

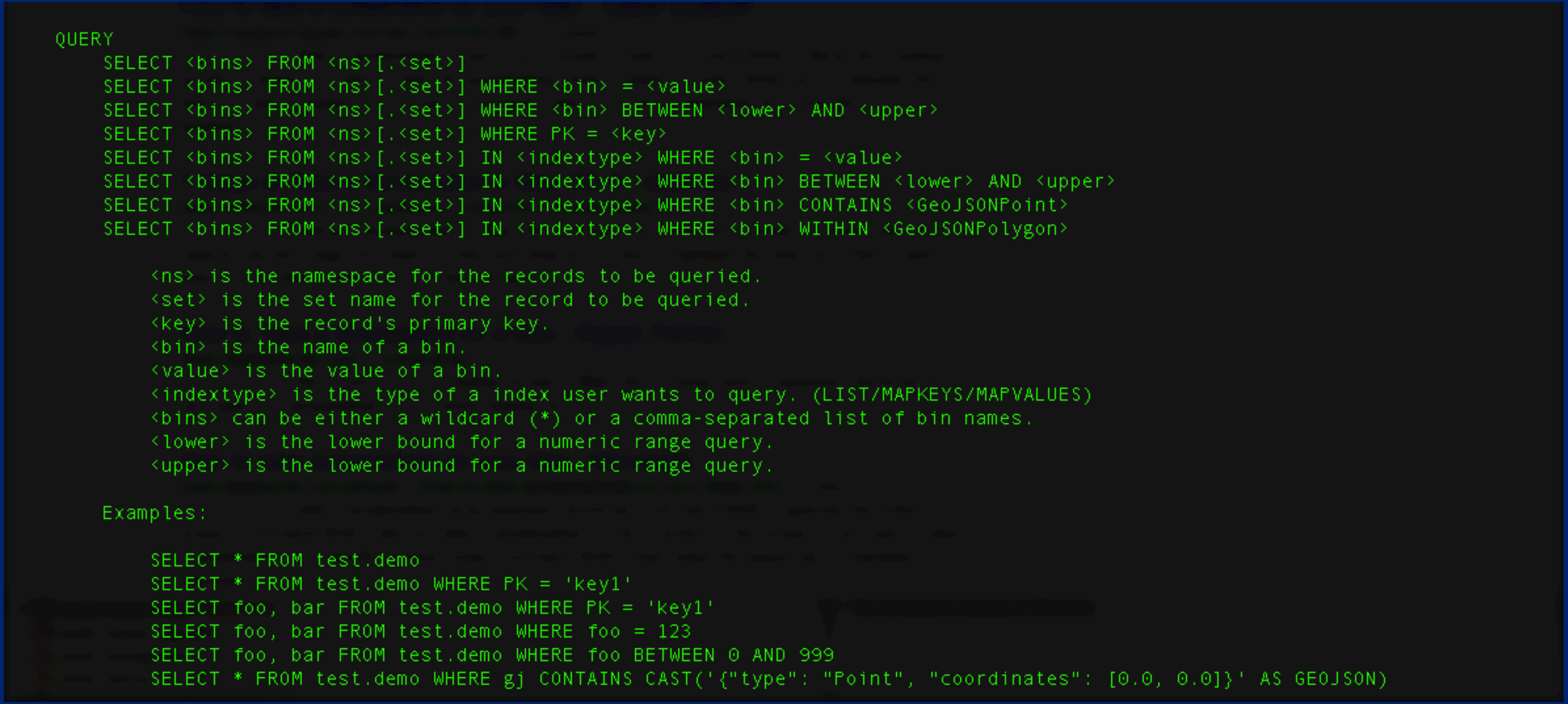


```
public void insertData(Status st) throws AerospikeException {
    WritePolicy pm = null;
    AerospikeClient client = null;

    try {
        client = new AerospikeClient("10.128.5.181", 3000);
        client.createIndex(null, "example", "tweet", "TestIndex","user_name", IndexType.STRING);

    } catch (AerospikeException e) {

        System.out.println("Connection Problem : " + e.getMessage());
    }

    Key key = new Key("example", "tweet", st.getId());

    Bin tweeID = new Bin("userID", st.getUser().getId());
    Bin userName = new Bin("user_name", st.getUser().getScreenName());
    client.put(pm, key, tweeID, userName);

    Bin retweetCount = new Bin("retweetcount", st.getRetweetCount());
    Bin userTimezone = new Bin("userTimeZone", st.getUser().getTimeZone());
    client.put(pm, key, retweetCount, userTimezone);

    Bin tweetText = new Bin("tweettext", st.getText());
    Bin FavoriteCount = new Bin("FavoriteCount", st.getUser().getFavouritesCount());
    client.put(pm, key, tweetText, FavoriteCount);

    Bin place = new Bin("Place", st.getPlace());
    Bin FollowerCount = new Bin("FollowerCount", st.getUser().getFollowersCount());
    client.put(pm, key, place, FollowerCount);

    client.close();

}
```



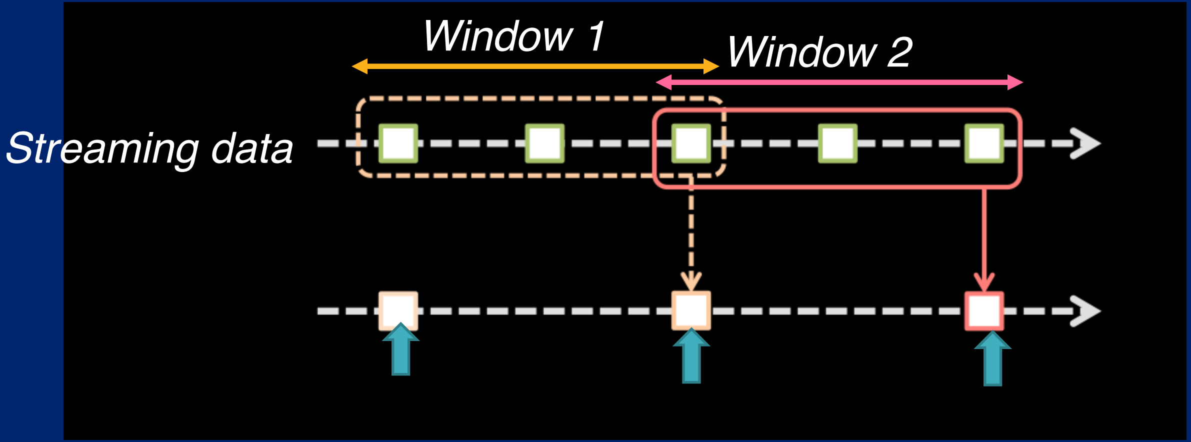
```
aql> select * from example
+--------------------+-------------------+--------------+------------------------------+-------------------------------------------
-------------------------------------------------------------------------------------------+---------------+---------------+
| userID             | user_name         | retweetcount | userTimeZone                 | tweettext
                                                                                           | FavoriteCount | FollowerCount |
+--------------------+-------------------+--------------+------------------------------+-------------------------------------------
-------------------------------------------------------------------------------------------+---------------+---------------+
| 753758311674212356 | "mrskaah_"        | 0            | "Pacific Time (US & Canada)" | "@mussuryG vou t socar amanhã"
                                                                                          | 1193          | 114           |
| 753758213640712192 | "nandaheriiiques" | 0            |                              | "As vzs tenho vontade de socar algumas pess
                                                                                           | 89            | 25            |
| 753758306674569216 | "zurdoting"       | 0            |                              | "RT @History_Futbol: NI EN EL FIFA. Un reco
                                                                                        | 153           | 57            |
| 753758315138707456 | "ZSportGuayana"   | 0            |                              | "RT @fifacom_es: At.Nacional luchará por la
                                                                                 | 68            | 446           |
| 753758302165663745 | "Adzy_bee"        | 0            |                              | "Check out my new FIFA vid!! Hope you enjoy
                                                                                           | 284           | 42            |
| 753758174465974273 | "weeklyjuice"     | 0            | "Mid-Atlantic"               | "RT @Shawnife_: Why do girls think its easy

Could lose 7 FIFA games in a row & still not tell my guy he's…"
| 753758294733357056 | "_adrianoomelo"   | 0            | "Mid-Atlantic"               | "Vontade de socar a amanda, puta que pariu
                                                                                           | 3229          | 920           |
| 753758324085194752 | "hobi1004"        | 0            | "Pacific Time (US & Canada)" | "RT @bts__imagine: amo tanto que quero soca
                                                                                           | 2378          | 116           |
| 753758016651063296 | "TequilaKass"     | 0            |                              | "Погба: «ФИФА стоит вручить «Золотой мяч» F
  | 314           |
| 753758210822025217 | "LucasJackson88"  | 0            |                              | "I just voted @AnthonyMartial to be on the
                                                                                           | 9             | 19            |
| 753758332876365825 | "jagg_17"         | 0            | "Caracas"                    | "Ranking mensual de la FIFA Clasificación:

46. Venezuela
51. Panamá
55. Jamaica
58. Trinidad y Tobago
82. Honduras
90. Guatemala"
| 753758296469798912 | "DareJiynx"       | 0            | "Eastern Time (US & Canada)" | "I liked a @YouTube video from @fazeblazike
```

Aerospike Query Language



```
QUERY
    SELECT <bins> FROM <ns>[.<set>]
    SELECT <bins> FROM <ns>[.<set>] WHERE <bin> = <value>
    SELECT <bins> FROM <ns>[.<set>] WHERE <bin> BETWEEN <lower> AND <upper>
    SELECT <bins> FROM <ns>[.<set>] WHERE PK = <key>
    SELECT <bins> FROM <ns>[.<set>] IN <indextype> WHERE <bin> = <value>
    SELECT <bins> FROM <ns>[.<set>] IN <indextype> WHERE <bin> BETWEEN <lower> AND <upper>
    SELECT <bins> FROM <ns>[.<set>] IN <indextype> WHERE <bin> CONTAINS <GeoJSONPoint>
    SELECT <bins> FROM <ns>[.<set>] IN <indextype> WHERE <bin> WITHIN <GeoJSONPolygon>

        <ns> is the namespace for the records to be queried.
        <set> is the set name for the record to be queried.
        <key> is the record's primary key.
        <bin> is the name of a bin.
        <value> is the value of a bin.
        <indextype> is the type of a index user wants to query. (LIST/MAPKEYS/MAPVALUES)
        <bins> can be either a wildcard (*) or a comma-separated list of bin names.
        <lower> is the lower bound for a numeric range query.
        <upper> is the lower bound for a numeric range query.

    Examples:

        SELECT * FROM test.demo
        SELECT * FROM test.demo WHERE PK = 'key1'
        SELECT foo, bar FROM test.demo WHERE PK = 'key1'
        SELECT foo, bar FROM test.demo WHERE foo = 123
        SELECT foo, bar FROM test.demo WHERE foo BETWEEN 0 AND 999
        SELECT * FROM test.demo WHERE gj CONTAINS CAST('{"type": "Point", "coordinates": [0.0, 0.0]}' AS GEOJSON)
```

Querying Fast Data



Select Distinct vehicleId
From PosSpeedStr [Range 30 Seconds]