

Creating and Querying Views

A *view* is a saved query, which then can be queried as if it were a table. The syntax for creating a view is the same as using CREATE TABLE AS SELECT (CTAS), but with VIEW instead of TABLE:

```
CREATE VIEW viewname AS
```

```
SELECT col1, col2, ... FROM tablename ... ;
```

Views appear in lists of the tables within a database (in Hue interfaces and in the result of SHOW TABLES) exactly as if they were tables.

For example, this query to show information about the aircraft flying into or out of a particular airport might be something you want to explore for different airports:

```
SELECT f.carrier, name, origin,  
       dest, type, manufacturer, model,  
       p.year, engines, seats, engine  
FROM flights f  
       JOIN airlines a  
       ON (f.carrier = a.carrier)  
       JOIN planes p
```

```
ON (f.tailnum = p.tailnum)
```

```
WHERE origin='BOS' OR dest='BOS';
```

Rather than running this every time you want to look at a different airport, you could save a view with all the information for all airports:

```
CREATE VIEW craft_information AS
```

```
SELECT f.carrier, name, origin,
```

```
dest, type, manufacturer, model,
```

```
p.year, engines, seats, engine
```

```
FROM flights f
```

```
JOIN airlines a
```

```
ON (f.carrier = a.carrier)
```

```
JOIN planes p
```

```
ON (f.tailnum = p.tailnum);
```

Then you can query the view:

```
SELECT * FROM craft_information WHERE origin='BOS' or dest='BOS';
```

Note that only the columns specified in the CREATE VIEW statement will be returned when you query the view. It's also possible to limit which rows can be returned by using a WHERE clause in the view definition. By limiting which columns and rows can be

returned, views can be used to prevent users from accessing sensitive information. So views can be used both for convenience *and* for security.

Try It!

1. Create the `craft_information` view described above.
2. Use `SHOW TABLES`; to get a list of tables in your active database. See if `craft_information` is included, and if it is, does it appear any differently from the actual tables in the list?
3. Use `DESCRIBE craft_information`; and then `DESCRIBE FORMATTED craft_information`; Note that there is nothing in the basic `DESCRIBE` results that indicates this is a view rather than a table, then find what there *is* in `DESCRIBE FORMATTED` that indicates this.
4. Run a query to return sample craft information for BOS (Logan International Airport in Boston), or another U.S. airport that you have flown from. (Not all airports will be included in the database, but you can certainly try! If you get 0 results, try a larger airport.) *Note:* Most airports are likely to return a large number of rows—BOS returns over 2 million rows, for example—so if you are using the command line, you should limit the number of rows returned. If you're interested, you could also limit the results by picking a particular carrier as well as the airport for your `WHERE` clause, but even then, you might get thousands of rows.
5. If this were a table, there would be a `craft_information` storage directory in the file system (for example, in `/user/hive/warehouse/` or `/user/hive/warehouse/fly.db`, depending on which database you had as your active database). Check HDFS for such a directory. (There should be none.) The view uses the same data as the source tables, so no storage directory is created.

Do not try dropping the view; you'll use it in the next reading.