

Topics in SQL

**ALY 6030 – Northeastern University
Week 4**

Rasoul Behboudi, PhD

Motivating questions

- How do we create variables for ranking using SQL?
- How can we store useful queries on the server?

Ranking in SQL, one approach:

```
SELECT *, @rownum := @rownum + 1 AS rank  
FROM Table, (SELECT @rownum := 0) r  
ORDER BY Standing [ASC, DESC];
```

- (SELECT @rownum := 0) initializes variable
- @local_variable: a local variable
- := assignment operator in MySQL

Ranking in SQL, example

- Example from edX lab – Rank the students in the Fall 2012 run of 6.002x:

```
SELECT userid_DI, grade, @rownum := @rownum + 1 AS rank
FROM course_users, (SELECT @rownum := 0) r
WHERE course_id = 'MITx/6.002x/2012_Fall'
ORDER BY grade DESC;
```

userid_DI	grade	rank
MHxPC130039429	1	24
MHxPC130073762	1	25
MHxPC130079236	1	26
MHxPC130564000	0.99	27
MHxPC130386800	0.99	28
MHxPC130366148	0.99	29

Views in SQL

- Named virtual table, acts as a stored query
- Data in views are stored in other tables, view is defined in terms of those tables
- When view is called in a query, will return up to date data, view is recreated each time the view is called (processed at run-time, can impact performance)
- Persists where database is stored (will still be there when you open Workbench), by default clients can see views of other clients
- Used for security so that people can only see subset of data (e.g. Marketing can only access these views, HR can access views related to payroll)
- Useful for input to dashboard (Tableau)

Views in SQL

```
CREATE VIEW VIEW_NAME AS  
SELECT colnames  
FROM tables  
WHERE conditions;
```

-- then:

```
SELECT * FROM VIEW_NAME;
```

```
DROP VIEW VIEW_NAME;
```

Views in SQL, example

```
SELECT a.state, a.city, a.city_pop, b.STATE_POP
FROM cities as a
INNER JOIN(
    SELECT state, sum(city_pop) as STATE_POP
FROM cities
GROUP BY state
) as b
ON a.state = b.state;
```

```
CREATE VIEW state_pop_vw AS
SELECT state, sum(city_pop) as state_pop
FROM cities
GROUP BY state;
```

```
SELECT a.state, a.city, a.city_pop, b.STATE_POP
FROM cities as a
INNER JOIN state_pop_vw as b
ON a.state = b.state;
```

Temporary tables in SQL

- Named tables for storing results
- Deleted when client session ends, frees up memory
- Users can create temporary tables with the same name without conflict with other users
- If temporary tables are created with the name of another table, that first table is hidden until end of session or temporary table is dropped

Temporary tables in SQL

```
CREATE TEMPORARY TABLE table_name (  
var1 VARCHAR NOT NULL,  
var2 INT NOT NULL DEFAULT 0);
```

```
INSERT INTO table_name  
VALUES ('value1', value2);
```

```
SELECT * FROM table_name;
```

Temporary tables in SQL

```
SELECT a.state, a.city, a.city_pop, b.STATE_POP
FROM cities as a
INNER JOIN(
    SELECT state, sum(city_pop) as STATE_POP
FROM cities
GROUP BY state
) as b
ON a.state = b.state;
```

```
CREATE TEMPORARY TABLE state_pop_tmp AS
SELECT state, sum(city_pop) as state_pop
FROM cities
GROUP BY state;
```

```
SELECT a.state, a.city, a.city_pop, b.STATE_POP
FROM cities as a
INNER JOIN state_pop_tmp as b
ON a.state = b.state;
```

Topics in SQL

- How can these skills help with data analysis?

