

# CSC 443 – Data Base Management Systems

## Lecture 7 – SQL As A Data Manipulation Language

### Objectives of SQL

- Ideally, database language should allow user to:
  - create the database and relation structures;
  - perform insertion, modification, deletion of data from relations;
  - perform simple and complex queries.
- Must perform these tasks with minimal user effort and command structure/syntax must be easy to learn.
- It must be portable.

## What is SQL (Structure Query Language)?

- SQL is a transform-oriented language with 2 major components:
  - A DDL for defining database structure.
  - A DML for retrieving and updating data.
- Until SQL:1999, SQL did not contain flow of control commands. These had to be implemented using a programming or job-control language, or interactively by the decisions of user.

## What is SQL? (continued)

- SQL is relatively easy to learn:
  - it is non-procedural - you specify *what* information you require, rather than *how* to get it;
  - it is essentially free-format.

## SELECT Statement

```
SELECT [DISTINCT | ALL]
      { * | [columnExpression [AS newName]] [,...] }
FROM      TableName [alias] [, ...]
[WHERE    condition]
[GROUP BY columnList] [HAVING condition]
[ORDER BY columnList]
```

5

Pearson Education © 2009

## SELECT Statement

FROM	Specifies table(s) to be used.
WHERE	Filters rows.
GROUP BY	Forms groups of rows with same column value.
HAVING	Filters groups subject to some condition.
SELECT	Specifies which columns are to appear in output.
ORDER BY	Specifies the order of the output.

## SELECT Statement

- Order of the clauses cannot be changed.
- Only SELECT and FROM are mandatory.

## SELECT - Example

```
mysql> select * from Staff;
```

```
+-----+-----+-----+-----+-----+-----+-----+
-----+-----+
| staffNo | fName | lName | position | sex | DOB      |
salary   | branchNo |
+-----+-----+-----+-----+-----+-----+-----+
-----+-----+
| SA9     | Mary  | Howe  | Assistant | f   | 1970-02-19 |
9000.00  | B007  |      |           |     |             |
| SG14    | David | Ford  | Supervisor | M   | 1958-03-24 |
18000.00 | B003  |      |           |     |             |
| SG37    | Ann   | Beech | Assistant  | f   | 1960-11-10 |
12000.00 | B003  |      |           |     |             |
| SG5     | Susan | Brand | Manager    | f   | 1940-06-03 |
24000.00 | B003  |      |           |     |             |
| SL21    | John  | White | Manager    | M   | 1945-10-01 |
30000.00 | B005  |      |           |     |             |
| SL41    | Julie | Lee   | Assistant  | f   | 1965-06-13 |
9000.00  | B005  |      |           |     |             |
+-----+-----+-----+-----+-----+-----+-----+
-----+-----+
```

```
6 rows in set (0.00 sec)
```

```
mysql> select staffNo, fName, lName from Staff;
```

```
+-----+-----+-----+
| staffNo | fName | lName |
+-----+-----+-----+
| SA9     | Mary  | Howe  |
| SG14    | David | Ford  |
| SG37    | Ann   | Beech |
| SG5     | Susan | Brand |
| SL21    | John  | White |
| SL41    | Julie | Lee   |
+-----+-----+-----+
```

```
6 rows in set (0.00 sec)
```

```
mysql>
```

## Use of **SELECT** - Examples

```
mysql> select * from Viewing;
```

```
+-----+-----+-----+-----+
| clientNo | propertyNo | viewDate | comment |
+-----+-----+-----+-----+
| CR56     | PA14       | 2008-05-24 | too small |
| CR56     | PG36       | 2008-04-28 | NULL     |
| CR56     | PG4        | 2008-05-26 | NULL     |
| CR62     | PA14       | 2008-05-14 | no dining room |
| CR76     | PG4        | 2008-04-20 | too remote |
+-----+-----+-----+-----+
```

```
5 rows in set (0.00 sec)
```

```
mysql> select propertyNo from Viewing;
+-----+
| propertyNo |
+-----+
| PA14      |
| PA14      |
| PG36      |
| PG4       |
| PG4       |
+-----+
5 rows in set (0.00 sec)
mysql> select distinct propertyNo from Viewing;
+-----+
| propertyNo |
+-----+
| PA14      |
| PG36      |
| PG4       |
+-----+
```

## Using a Calculated Field - Example

```
mysql> select staffNo, fName, lName, salary from Staff;
+-----+-----+-----+-----+
| staffNo | fName | lName | salary |
+-----+-----+-----+-----+
| SA9     | Mary  | Howe  | 9000.00 |
| SG14    | David | Ford  | 18000.00 |
| SG37    | Ann   | Beech | 12000.00 |
| SG5     | Susan | Brand | 24000.00 |
| SL21    | John  | White | 30000.00 |
| SL41    | Julie | Lee   | 9000.00 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

```
mysql> select staffNo, fName, lName, salary/12 from Staff;
```

```
+-----+-----+-----+-----+
| staffNo | fName | lName | salary/12 |
+-----+-----+-----+-----+
| SA9     | Mary  | Howe  | 750.000000 |
| SG14    | David | Ford  | 1500.000000 |
| SG37    | Ann   | Beech | 1000.000000 |
| SG5     | Susan | Brand | 2000.000000 |
| SL21    | John  | White | 2500.000000 |
| SL41    | Julie | Lee   | 750.000000 |
+-----+-----+-----+-----+
```

```
6 rows in set (0.01 sec)
```

```
mysql>
```

## Formatting a Calculated Field

```
mysql> select staffNo, fName, lName, CAST(salary / 12 as
decimal(9, 2)) as monthlySalary from Staff;
```

```
+-----+-----+-----+-----+
| staffNo | fName | lName | monthlySalary |
+-----+-----+-----+-----+
| SA9     | Mary  | Howe  |          750.00 |
| SG14    | David | Ford  |         1500.00 |
| SG37    | Ann   | Beech |         1000.00 |
| SG5     | Susan | Brand |         2000.00 |
| SL21    | John  | White |         2500.00 |
| SL41    | Julie | Lee   |          750.00 |
+-----+-----+-----+-----+
```

```
6 rows in set (0.00 sec)
```

```
mysql>
```

## SELECT with Row Selection - Example

```
mysql> select staffNo, fName, lName, position, salary
-> from Staff
-> where salary > 10000;
```

staffNo	fName	lName	position	salary
SG14	David	Ford	Supervisor	18000.00
SG37	Ann	Beech	Assistant	12000.00
SG5	Susan	Brand	Manager	24000.00
SL21	John	White	Manager	30000.00

4 rows in set (0.00 sec)

```
mysql>
```

```
mysql> select *
-> from Branch
-> where City = 'London' or city = 'Glasgow';
```

branchNo	street	city	postcode
B002	56 Clover Dr	London	NW10 6EU
B003	163 Main St	Glasgow	G11 9QX
B005	22 Deer Rd	London	SW1 4EH

3 rows in set (0.00 sec)

```
mysql>
```



```
mysql> select staffNo, fName, lName, position, salary
-> from Staff
-> where salary between 20000 and 30000;
```

```
+-----+-----+-----+-----+-----+
| staffNo | fName | lName | position | salary |
+-----+-----+-----+-----+-----+
| SG5     | Susan | Brand | Manager  | 24000.00 |
| SL21    | John  | White | Manager  | 30000.00 |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

```
mysql>
```

```
mysql> select staffNo, fName, lName, position, salary
-> from Staff
-> where salary >= 20000 and salary <= 30000;
```

```
+-----+-----+-----+-----+-----+
| staffNo | fName | lName | position | salary |
+-----+-----+-----+-----+-----+
| SG5     | Susan | Brand | Manager  | 24000.00 |
| SL21    | John  | White | Manager  | 30000.00 |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

```
mysql>
```

```
mysql> select staffNo, fName, lName, position
-> from Staff
-> where position in ('Supervisor', 'Manager');
```

staffNo	fName	lName	position
SG14	David	Ford	Supervisor
SG5	Susan	Brand	Manager
SL21	John	White	Manager

```
3 rows in set (0.00 sec)
```

```
mysql>
```

```
mysql> select staffNo, fName, lName, position
-> from Staff
-> where position in ('Manager', 'Supervisor')
-> order by lName, fName;
```

staffNo	fName	lName	position
SG5	Susan	Brand	Manager
SG14	David	Ford	Supervisor
SL21	John	White	Manager

```
3 rows in set (0.00 sec)
```

```
mysql>
```

## SELECT and Pattern Matching

```
mysql> select ownerNO, fName, lName, address, telNo
-> from PrivateOwner
-> where address like '%Glasgow%';
```

ownerNO	fName	lName	address	telNo
CO40	Tina	Murphy	63 Well St, Glasgow G42	0141-943-1728
CO87	Carol	Farrel	6 Achray St, Glasgow G32 9DXX	0141-357-7419
CO93	Tony	Shaw	12 Park Place, Glasgow G4 0QR	0141-225-7025

```
3 rows in set (0.00 sec)
```

## SELECT and NULL Search

```
mysql> select clientNo, viewDate
-> from Viewing
-> where propertyNo = 'PG4' and comment is NULL;
```

clientNo	viewDate
CR56	2008-05-26

```
1 row in set (0.00 sec)
```

```
mysql>
```

## SELECT and Sorting

```
mysql> select staffNo, fName, lName, salary
       -> from Staff
       -> Order BY salary DESC;
```

```
+-----+-----+-----+-----+
| staffNo | fName | lName | salary |
+-----+-----+-----+-----+
| SL21    | John  | White | 30000.00 |
| SG5     | Susan | Brand | 24000.00 |
| SG14    | David | Ford  | 18000.00 |
| SG37    | Ann   | Beech | 12000.00 |
| SA9     | Mary  | Howe  | 9000.00  |
| SL41    | Julie | Lee   | 9000.00  |
+-----+-----+-----+-----+
```

6 rows in set (0.00 sec)

```
mysql>
```

## SELECT and Sorting

```
mysql> select propertyNo, type, rooms, rent
       -> from PropertyForRent
       -> Order by type DESC;
```

```
+-----+-----+-----+-----+
| propertyNo | type  | rooms | rent |
+-----+-----+-----+-----+
| PA14       | House | 6     | 650  |
| PG21       | House | 5     | 600  |
| PG16       | Flat  | 4     | 450  |
| PG36       | Flat  | 3     | 375  |
| PG4        | Flat  | 3     | 350  |
| PL94       | Flat  | 4     | 400  |
+-----+-----+-----+-----+
```

6 rows in set (0.00 sec)

```
mysql>
```

## SELECT and Sorting

```
mysql> select propertyNo, type, rooms, rent
      -> from PropertyForRent
      -> Order by type, rent DESC;
```

propertyNo	type	rooms	rent
PG16	Flat	4	450
PL94	Flat	4	400
PG36	Flat	3	375
PG4	Flat	3	350
PA14	House	6	650
PG21	House	5	600

```
6 rows in set (0.00 sec)
```

```
mysql>
```

## Using SQL Aggregate Functions

```
mysql> select count(*) as myCount
      -> from PropertyForRent
      -> where rent > 0;
```

myCount
6

```
1 row in set (0.00 sec)
```

```
mysql>
```

## Using SQL Aggregate Functions

```
mysql> select count(distinct propertyNo) as myCount
-> from Viewing
-> where viewDate Between '2008-05-01' and '2008-05-31';
```

```
+-----+
| myCount |
+-----+
|      2 |
+-----+
1 row in set (0.04 sec)
```

```
mysql>
```

## Using SQL Aggregate Functions

```
mysql> select count(staffNo) as myCount, sum(salary) as
mySum
-> from Staff
-> where position = 'Manager';
```

```
+-----+-----+
| myCount | mySum   |
+-----+-----+
|      2 | 54000.00 |
+-----+-----+
1 row in set (0.00 sec)
```

```
mysql>
```

## Using SQL Aggregate Functions

```
mysql> select min(salary) as myMin, max(salary) as myMax,  
-> CAST(avg(salary) as  
-> decimal(9, 2)) as myAvg  
-> from Staff;
```

```
+-----+-----+-----+  
| myMin  | myMax  | myAvg  |  
+-----+-----+-----+  
| 9000.00 | 30000.00 | 17000.00 |  
+-----+-----+-----+  
1 row in set (0.00 sec)
```

```
mysql>
```

## Restricted Grouping Using Having

```
mysql> select branchNo, count(staffNo) as myCount,  
-> sum(salary) as mySum  
-> from Staff  
-> group by branchNo  
-> having count(staffNo) > 1  
-> order by branchNo;
```

```
+-----+-----+-----+  
| branchNo | myCount | mySum  |  
+-----+-----+-----+  
| B003     | 3       | 54000.00 |  
| B005     | 2       | 39000.00 |  
+-----+-----+-----+  
2 rows in set (0.00 sec)
```

```
mysql>
```