LEARNING OBJECTIVES

On completion of this session you will be able to:

- Use aggregate and SQL functions
- Manipulate sets of data
- Write subqueries
- Manipulate data in the database
Reading

Prescribed readings


Further references

Websites:


http://sqlzoo.net/ A Gentle Introduction to SQL

http://www.sqlcourse.com/ SQLCourse.com

http://www.1keydata.com/sql/sql.html SQL Tutorials

Where we are

```
+--------------------------+
| Introduction to Database |
| Systems                  |
| The Relational Model     |
+--------------------------+
| Database Lifecycle       |
| Conceptual Design        |
| Logical Design           |
| Physical Design          |
| Normalisation            |
|                          |
| Implementation           |
| SQL (DML)                |
| SQL (DDL & DCL)          |
| Transaction Management   |
| Database Administration  |
| Data Warehousing, Data   |
| Mining, ECommerce        |
+--------------------------+
```
1. Introduction

In the previous study guide we learned the SQL data manipulation command, SELECT which is used to retrieve data from the database. In this study guide we build on what you learned in Study Guide 7 and learn how to use additional SELECT clauses and manipulate data in the database.

2. Using functions

2.1 Aggregate functions

Are used to perform mathematical summaries such as, counting the number of rows, finding the minimum and maximum values for some specified attribute, summing the values in a column, and averaging the values in a specified column.

<table>
<thead>
<tr>
<th>Read</th>
<th>R &amp; C 6th Edn, p.276 – 280 Section 6.6.3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R &amp; C 7th Edn, p.255 – 259 Section 7.6.3</td>
</tr>
</tbody>
</table>

2.2 SQL functions

SQL functions are useful to manipulate data by decomposing the data elements. Functions use numerical, date or string values and may appear in a SQL statement wherever a value or attribute is used. The functions supported by the DBMSs differ between vendors.

<table>
<thead>
<tr>
<th>Read</th>
<th>R &amp; C 6th Edn, p. 347 – 353 Section 7.4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R &amp; C 7th Edn, p. 312 – 317 Section 8.4</td>
</tr>
</tbody>
</table>
3. Manipulating sets of data

3.1 GROUP BY clause

The GROUP BY clause is used to divide the rows in a table into groups. You can then use the aggregate functions to return summary information for each group.

When using the GROUP BY clause, make sure that all columns in the SELECT list that are not in the group functions are included in the GROUP BY clause.

The GROUP BY column does not have to be in the SELECT clause.

Read
R & C 6th Edn, p.280 – 284 Section 6.6.4
R & C 7th Edn, p.260 – 262 Section 7.6.4

3.2 Relational SET operators

SQL data manipulation commands are set oriented, that is they operate over entire sets of rows and columns at once. Using the set operators you can combine two or more sets to create new sets (relations).

Read
R & C 6th Edn, p.320 Section 7.1
R & C 7th Edn, p.286 Section 8.1

UNION

The UNION statement combines rows from two or more queries without including duplicate rows.

The UNION ALL statement combines rows from two or more queries and retains the duplicate rows.

Read
R & C 6th Edn, p. 321 – 323 Section 7.1.1 – 7.1.2
R & C 7th Edn, p. 286 – 288 Section 8.1.1 – 8.1.2
The INTERSECT statement combines rows from two queries and returns only those rows that appear in both sets.

Read  
R & C 6th Edn, p. 323 – 324  Section 7.1.3  
R & C 7th Edn, p. 288 – 290  Section 8.1.3

The MINUS statement combines rows from two queries and returns only those rows that appear in the first set but not in the second.

Read  
R & C 6th Edn, p.324 - 325  Section 7.1.4  
R & C 7th Edn, p.289 - 290  Section 8.1.4

4. HAVING clause

The HAVING clause is applied to the output of a GROUP BY operation to restrict the selected rows.

The HAVING clause operates like a WHERE clause, however, the WHERE clause applies to columns and expressions for individual rows, while the HAVING clause is applied to the output of a GROUP BY operation.

It is important to understand how the DBMS evaluates the clauses in a SQL statement.

The order of evaluation is as follows:

1. WHERE clause  
2. GROUP BY clause  
3. HAVING clause

Therefore if we wish to restrict the results of a query based on the result of a GROUP BY clause we need to use a HAVING clause rather than the WHERE clause.
5. **Subqueries**

A subquery is a query that is embedded (or nested) inside another query. Also known as nested query or an inner query.

Syntax:

```
SELECT select_list
FROM table
WHERE expr operator (SELECT select_list
FROM table);
```

The first query in the SQL statement is known as the outer query.

The query inside the SQL statement is known as the inner query.

The inner query is evaluated first and the output from this query is used as the input for the outer query.

The inner query is normally expressed inside parentheses.

6. **Manipulating data**

6.1 **INSERT**

The INSERT statement is used to enter data into a table.

```
INSERT INTO table [(column [, column...])] VALUES (value [, value...]);
```

If you insert a new row that contains values for each column in the table, the column list is not required in the INSERT clause. However, if you do not use the column list, the values must be listed according to the default order of the columns in the table.

The INSERT statement allows the insertion of data one row at a time. To insert multiple rows of data use an INSERT statement with a subquery.

```
INSERT INTO table [ column (, column ) ]
(subquery);
```
6.2 UPDATE

The UPDATE statement allows you to change attribute values in one or more rows of a table.

```
UPDATE table SET column = value [, column = value, ...] 
WHERE condition;
```

You can use subqueries in an UPDATE statement to update rows in a table based on values from another table.

```
UPDATE table SET column = (subquery) 
WHERE condition; 
```

6.3 DELETE

The DELETE statement allows you to delete rows of data from a table.

```
DELETE FROM table 
WHERE condition;
```

You can use subqueries in a DELETE statement to delete rows in a table based on values from another table.

```
DELETE FROM table 
WHERE condition = (subquery); 
```
7. Summary

In this study guide we built on what we learned in Study Guide 7 and learnt how to use additional SELECT clauses and manipulate data in the database. We have learned how the aggregate functions perform arithmetic computations over a set of rows. The aggregate functions are usually used in conjunction with the GROUP BY clause to group the output of aggregate functions by one or more attributes. The HAVING clause is used to restrict the output of the GROUP BY clause by selecting the aggregate rows that match a given condition. We also looked at the SQL functions that are used to extract or transform data. We have learnt how subqueries are used to evaluate data based on data from another query. We also learnt about the DML commands INSERT, UPDATE and DELETE that allow you to add, modify and delete data in a table.