

CHAPTER 4

Data Manipulation Language (DML) – Selecting Columns

4.1 Data Manipulation Language

The DML component of SQL comprises 4 basic statements:

SELECT	to retrieve rows from tables
UPDATE	to modify the rows of tables
DELETE	to remove rows from tables
INSERT	to add new rows to tables

4.2 The SELECT Statement

Chapters 4 to 8 illustrate with examples the data retrieval capabilities of the SELECT statement as well as introducing more general features of SQL such as arithmetic and logical expressions that apply equally to the UPDATE, DELETE and INSERT statements that are dealt with in later chapters.

Each chapter contains examples illustrating the various options available with each statement. You are advised to try out these examples for yourself. In addition, at the end of each chapter there are a number of exercise queries designed to test your understanding of the chapter.

The simplest form of the SELECT statement consists of just two clauses:

```
SELECT column-list
FROM table-list
```

4.2.1 Specifying the Required Column List

Example 4.2.1(a) – *List the full details of all Aircraft*

In order to list all the details (columns) recorded for each aircraft type we list the name of each column in the AIRCRAFT table as follows:

```
SELECT AIRCRAFT_TYPE, AIRCRAFT_DESCRIPTION,
NUMBER_OF_SEATS
FROM AIRCRAFT
```

Result:

AIRCRAFT_TYPE	AIRCRAFT_DESCRIPTION	NUMBER_OF_SEATS
737	Boeing 737-300 Jet	300
ATP	Advanced Turboprop	48
DC9	McDonnell Douglas Jet	120

For those queries in which all the columns of a particular table are to be retrieved, the column-list may be replaced by an asterisk. Hence:

```
SELECT *
FROM AIRCRAFT
```

gives the same result

Example 4.2.1(b) – List the names and addresses of all passengers

This query requires the selection of individual columns from the Passenger table, and so we simply list the columns required as follows:

```
SELECT NAME, ADDRESS
FROM PASSENGER
```

Result:

NAME	ADDRESS
D N Hamer	1 St Paul's Churchyard
D E Avison	5 Chancery Lane
G B Davis	25 Allenby Road
C Evans	63 Kew Green
J Millar	Englewood Cliffs
J D Ullman	1 Microsoft Way
A Smithson	16 Bedford St
D Etheridge	4 Marylands Avenue
E Simon	8 Cherry Street
A N Smith	81 Digby Crescent
T Pittman	The Little House
J Peters	31 Lucas Road
K E Kendall	11 Rosedale Avenue
R H Miller	155 Kingston Road

4.2.2 Removing Duplicate Rows

Example 4.2.2(a) – List all Flight Numbers with tickets issued against them

Flights for which tickets have been issued are recorded in the ITINERARY table. The following simple query lists FLIGHT_NOs from the ITINERARY table:

```
SELECT FLIGHT_NO
FROM ITINERARY
```

Result:

FLIGHT_NO
BD80
BD95
BD80
BD95
BD82
BD54
BD652
BD655
BD775
BD776
BD652
BD655
BD772
BD412
BD419
BD412
BD419
BD224
BD255
BD256
BD275
BD412
BD582
BD589
BD332
BD51
BD54
BD658
BD659
BD771
BD774

(31 rows are retrieved)

However, a given FLIGHT_NO will appear as many times as the number of tickets issued for that flight, and therefore the result of this query contains duplicate rows.

Example 4.2.2(b) – *List all Flight Numbers with tickets issued against them, removing duplicate rows*

To remove duplicates from the result of this sort of query SQL provides the DISTINCT function, which is used as follows:

```
SELECT DISTINCT FLIGHT_NO
FROM ITINERARY
```

Result:

FLIGHT_NO
BD224
BD255
BD256
BD275
BD332
BD412
BD419
BD51
BD54
BD582
BD589
BD652
BD655
BD658
BD659
BD771
BD772
BD774
BD775
BD776
BD80
BD82
BD95

(23 rows are retrieved)

4.2.3 Arithmetic Expressions

SQL allows arithmetic expressions to be included in the SELECT clause. An arithmetic expression consists of a number of column names and values connected by any of the following operations:

- + Add
- Subtract
- * Multiply
- / Divide

When included in the SELECT clause the results of an expression are displayed as a calculated table column.

Example 4.2.3 – Assuming Tariffs are recorded excluding VAT, calculate and list Tariffs inclusive of VAT.

There are a number of different ways of calculating Fare prices inclusive of VAT at 17.5%. The following are all equivalent and valid arithmetic expressions for calculating VAT-inclusive values within the SELECT clause:

- 1: Price + Price & .175
- 2: Price + Price * 17.5 / 100
- 3: Price * 1.175

```
SELECT ROUTE_NO, FARE_TYPE, PRICE, PRICE*1.175 AS
VAT_PRICE
FROM TARIFF
```

Result:

ROUTE_NO	FARE_TYPE	PRICE	VAT_PRICE
3	BUR	£117.00	£137.48
3	SDR	£158.00	£185.65
3	SDS	£79.00	£92.83
4	SBS	£49.00	£57.58
4	SDR	£162.00	£190.35
6	BUR	£117.00	£137.48
6	KFS	£53.00	£62.28
6	SBS	£42.00	£49.35
7	SDR	£128.00	£150.40
8	SDS	£74.00	£86.95
9	APR	£95.00	£111.63
9	EUR	£181.00	£212.68
9	PXR	£153.00	£179.78
11	KFS	£59.00	£69.33
13	EXR	£121.00	£142.18
14	SBS	£33.00	£38.78
14	SDR	£110.00	£129.25
15	SBS	£33.00	£38.78

4.2.4 Aggregate (Group) Functions

SQL provides a number of special functions called aggregate functions, which may be included in the SELECT clause. Each of these functions operates over a named column, and returns a *single value* calculated from all the selected rows.

AVG(<i>column-name</i>)	returns the average value
COUNT(<i>column-name</i>)	returns the number of non-null values
COUNT(*)	returns the number of rows
MAX(<i>column-name</i>)	returns the highest value
MIN(<i>column-name</i>)	returns the lowest value
SUM(<i>column-name</i>)	calculates the total of values

Example 4.2.4(a) – List the average aircraft seating capacity

```
SELECT AVG(NUMBER_OF_SEATS)
FROM AIRCRAFT
```

Result:

AVG(NUMBER_OF_SEATS)
156

Example 4.2.4(b) – Count the number of routes flown by the airline

```
SELECT COUNT(*)
FROM ROUTE
```

Result:

COUNT(*)
10

Aggregate functions return a single valued result, i.e. a calculated column with only one row. As such these functions cannot be included in the SELECT clause with columns that are multi-valued – i.e. returning two or more rows. On this basis **the following query is illegal**, and would not be processed by the DBMS:

```
SELECT AIRCRAFT_TYPE, AVG(NUMBER_OF_SEATS)
FROM AIRCRAFT
```

EXERCISE 3

Give the SQL required to:

1. List the full details of all airports
2. List the distinct AIRCRAFT_TYPES assigned to flights
3. List all tickets, with TICKET_DATE appearing in the leftmost column
4. From amongst all of the different aircraft types find the largest seating capacity. *(Note: you are not asked to find which particular aircraft type this seating capacity belongs to).*

Format the query given in example 4.2.3, so that the expression **price*1.175** is given a column heading of **vat_price**