# SETS and SQL

This exercise will take you through creating three sets of employees from the EMP table in the employee database. For a number of expressions, you will then be asked to

- evaluate each expression on paper and write out the result set
- express the set notation in everyday English
- convert the English expressions into an SQL query

You will be working with the ENAME field from the EMP table in these exercises

## Task 1 Creating the subsets

In maths, we frequently make up distinct names for items or collections of items (the term *distinct* is sometimes used for *unique*). For example – imagine a set of the department names, which we could call DEPTNM –

DEPTNM – the set of department names

This would be the set { Accounting, Research, Sales, Operations }

Open the *employee* database in Access and use the EMP table to write down (on paper) the following sets – make sure you use the {} notation correctly.

UN – the universal set of employee names (all the names in the EMP table)

DN1 – the names of employees that work in department 1

SAL24 – the names of employees whose salary is less than £24000

MGR875 – the names of employees with manager number 875

## Task 2 Creating the SQL for the sets in task 1

We can write SQL expressions to show the sets in task 1. For example

DEPTNM – { Accounting, Research, Sales, Operations }

Could be generated by

```
select dname
from department
```

Create four SQL queries that will create the sets in Task 1. You will need to use the SELECT – FROM – WHERE statement, with combinations of the *projection* and *selection* operation.

## Task 3 Working out various sub sets

To evaluate an expression containing sets, work out the resulting set when the operations are applied. For example

DN1 ∩ SAL24 would be

{ Pollard, Parker, Black } ∩ { March, Bell, Ahmad, Hayes, Cassy, Black }

Which would give {Black}

Evaluate the following expressions (write the resulting sets down on paper)

1. DN1 ∪ SAL24
2. DN1 ∩ MGR875
3. SAL24 ∪ SAL24$^{c}$
4. MGR875 – SAL24
5. ( DN1 ∩ MGR875 )$^{c}$
6. DN1$^{c}$ ∪ MGR875$^{c}$
7. ( SAL24 – DN1 ) – MGR875
8. SAL24 – ( DN1 – MGR875 )

## Task 4 Explaining the set in everyday terms

Example

DN1 ∩ SAL24 gives the resulting set { Black }

In English this is “the set containing people who work in department 1 and earn less than 24000”

For each of the resulting sets in task 3, write down the expression in everyday English

## Task 5 Expressing the sets in SQL

Example

DN1 ∩ SAL24 is “the set containing people who work in department 1 and earn less than 24000”. In SQL this would be

```
select ename
from emp
where deptno=1 and sal<24000
```

Write down the SQL for each of the everyday expressions from Task 4. You can use Access to test out and refine your SQL solutions.

## Extra: Implementing SET operations in SQL

SQL does offer a UNION keyword, so you can write

```
select ename from emp where sal<24000
UNION
select ename from emp where mgrno=875
```

You have to have what SQL calls "union compatibility" (the number and type of columns in the two select statements must be aligned).

There is no MINUS, but probably the best way to do it is to use NOT IN e.g. SAL24 - MGR875 would be

```
SELECT ename FROM emp where sal<24000 and ename NOT IN
(SELECT ename FROM emp where mgr = 875)
```

Unfortunately there is no INTERSECTION either, which means either using a rather clunky join or using the IN keyword, e.g. SAL24 ∩ DEPTNO1 would be

```
SELECT ename FROM `emp` where sal<24000 and ename IN
(SELECT ename FROM emp where deptno=1)
```