*Sheffield* *Hallam* *University*



Faculty of Science, Technology and Arts

**SQL** **WORKBOOK**

**55-500998**

**Database** **Systems** **For**

**Software** **Applications**

**SECTION G**

**2020/2021**



*Sheffield* *Hallam* *University*

Faculty of Science, Technology and Arts

**STRUCTURED QUERY LANGUAGE** **(SQL)**

**2020/2021**

SQL Workbook 1 September 2020

SQL Workbook 2 September 2020

Introduction

**INTRODUCTION**

The purpose of this book is to provide practical exercises in the use of SQL to create, populate and maintain a relational database.

**SQL**

SQL (Structured Query Language) is an ISO and ANSI standard *database* *query* *language.* Most relation databases are SQL-compliant, but in spite of SQL being a standard, SQL code is not completely portable among different database management systems.

**Oracle Relational Database**

For this module we will be using Oracle Database 11g Enterprise Edition.

Students will need to create a user account for Oracle, - how to do this is shown in Appendix X. If you have an account from a previous module, it will still be valid – to reset the password or unlock an account, also see Appendix X

**Oracle SQL Developer**

Students will need to connect to their Oracle database account using SQL Developer which is a **free** graphical tool for database development. How to do this is shown in Appendix Y.

With SQL Developer, you can create, browse and manage database objects, execute SQL statements and SQL scripts, and import, manipulate, and export data.

**Sample Tables**

All new accounts have a set of default tables (EMP, DEPT and SALGRADE) already created and populated with data. These are referred to as the **PERSONNEL** **SYSTEM**. Many of the **exercises** throughout this workbook are based on the Personnel System tables.

Some lecture material and various **worked** **examples** in this text are based on part of a simple **BANK** **ACCOUNTING** **SYSTEM** and use the tables CUST, CUSTACC and ACC. These tables represent the fact that a bank customer may have many accounts, and that an account may be held jointly by more than one customer.

Details of these tables can be found in the following pages. If your Personnel System tables get changed/corrupted the instructions to delete and recreate them can be found at the bottom of page 5.

**Workflow**

Topics should be tackled in sequence. This is because exercises in later sections may rely on changes you make to your data or data structures in earlier sections to work properly. Topics must be completed before the scheduled session of the next topic.

SQL Workbook 3 September 2020

Introduction

**The Bank Accounting System**

CUST Owns CUSTACC Allocated ACC

Table: **CUST**

|  |  |  |  |
| --- | --- | --- | --- |
| REFNO | NAME | ADDRESS | AREA |
| A123 A124 B127 B128  C371 | J Doe J Smith R Best J Best  R Done | 1 High Street 2 West Street 4 East Row  4 East Row  23 Middle Avenue | Sheffield Sheffield Rotherham Rotherham  Barnsley |

Table: **CUSTACC**

|  |  |
| --- | --- |
| REFNO | ACCNO |
| A123 A123 B127  B128 | 1245890 1494315 5418490  5418490 |

Table: **ACC**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ACCNO | BALANCE | BRANCH | OPENED | BONUS |
| 1245890 1494315  5418490 | 234.50 0.50  1789.40 | Broomhill Tinsley  Broomhill | 12 Nov 2003 15 Dec 1999  6 May 1988 | 100.00 0.00 |

SQL Workbook 4 September 2020

Introduction

**THE PERSONNEL SYSTEM**

Table: **EMP**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| EMPNO | ENAME | JOB | MGR | HIREDATE | SAL | COMM | DEPTNO |
| 7369 7499 7521 7566 7654 7698 7782 7788 7839 7844 7876 7900 7902  7934 | SMITH ALLEN WARD JONES MARTIN BLAKE CLARK SCOTT KING TURNER ADAMS JAMES FORD  MILLER | CLERK SALESMAN SALESMAN MANAGER SALESMAN MANAGER MANAGER ANALYST PRESIDENT SALESMAN CLERK CLERK ANALYST  CLERK | 7902 7698 7698 7839 7698 7839 7839 7566  7698 7788 7698 7566  7782 | 17-DEC-80 20-FEB-81 22-FEB-81 02-APR-81 28-SEP-81 01-MAY-81 09-JUN-81 09-DEC-82 17-NOV-81 08-SEP-81 12-JAN-83 03-DEC-81 03-DEC-81  23-JAN-82 | 800.00 1600.00 1250.00 2975.00 1200.00 2850.00 2450.00 3000.00 5000.00 1500.00 1100.00 950.00 3000.00  1300.00 | 300.00 500.00  1250.00  0.00 | 20 30 30 20 30 30 10 20  30 20 30 20  10 |

Table: **DEPT**

|  |  |  |
| --- | --- | --- |
| DEPTNO | DNAME | LOC |
| 10 20 30 40 | ACCOUNTING RESEARCH SALES OPERATIONS | NEW YORK DALLAS CHICAGO BOSTON |

Table: **SALGRADE**

|  |  |  |
| --- | --- | --- |
| GRADE | LOSAL | HISAL |
| 1 2 3 4  5 | 700.00 1201.00 1401.00 2001.00  3001.00 | 1200.00 1400.00 2000.00 3000.00  9999.00 |

Should data in the tables become corrupt, they may be restored to their original status by issuing each of the following statements for the appropriate table:

**DROP** **TABLE** EMP ;

**CREATE** **TABLE** EMP **AS** **SELECT** \* **FROM** EXAMPLE.EMP ;

SQL Workbook 5 September 2020

Introduction

This page is intentionally blank.

SQL Workbook 6 September 2020

Section G

**NESTED SUBQUERIES**

**NESTED SUBQUERIES**

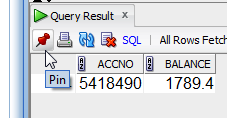
Some questions are difficult to resolve in one pass. For example, if we need to find out which bank account has the largest balance, it cannot be done in one simple select query. We can calculate MAX(balance), but we won’t know which account it is.

But if we know what the value of MAX(balance) is, a second query can be used to find the account with that balance.

The solution to problems of this type lies is the sub-query – a powerful feature of SQL.

Any expression in a SQL statement can be replaced by a subquery. The subquery is then processed first, and the result of the subquery is substituted into the original statement.

**Example**



**SELECT** ACCNO, BALANCE **FROM** ACC

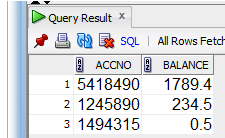
**WHERE** BALANCE **=**

(**SELECT** MAX(BALANCE) **FROM** ACC) ;

In this example the inner statement (subquery) will return a single value (1,789.40). That value is then tested against the value in the balance field of each of the rows in the ACC table. Every row with that value will be returned in the result table.

Care needs to be taken with the WHERE clause. In the example above, the nested SELECT query returns a single value and so the ‘**WHERE** BALANCE **=**’ predicate is valid.

But if the inner SELECT can return several rows then the condition would need to take this into account e.g. **WHERE** column **IN** (**SELECT**…..).

**Example**

**SELECT** ACCNO, BALANCE **FROM** ACC

**WHERE** BRANCH

**IN** (**SELECT** DISTINCT BRANCH **FROM** ACC);

**Remember**, the subquery is a complete query in its own right, it should be written first, run and checked, and only then incorporated into the full query.

SQL Workbook 63 September 2020

Section G

**Exercises**

**G1** Who has the highest basic salary?

This cannot be answered straight off - try finding the answer by looking at the printed tables from the Personnel System. Firstly, we need to know what is the highest salary (and that can be only one value) before we can decide who has that salary - and there may be more than one employee with that salary.

What is the highest salary?

**SELECT** MAX(SAL) FROM EMP;

Run the query. How many results are displayed? There should be only one!

Now place that Select statement as the sub-query (enclosed in brackets and without the semi-colon) and determine who has that salary.

**SELECT** ENAME,SAL **FROM EMP WHERE SAL IN (SELECT MAX(SAL) FROM EMP);**

**G2** Which salesman earns the most, including commission?

Hint - Create and check a query to determine the highest income (salary + commission) for salesmen, and then determine which salesmen earn that income by using that query as the subquery.

**SELECT ENAME, (SAL + NVL(COMM, 0)) AS INCOME**

**FROM EMP**

**WHERE (SAL + NVL(COMM, 0)) =**

**(SELECT MAX(SAL + NVL(COMM, 0)) AS INCOME**

**FROM EMP**

**WHERE JOB = 'SALESMAN');**

Is CLARKE one of the displayed employees? He is a MANAGER!

SQL Workbook 64 September 2020

Section G

**G3** Who in New York has the highest income (salary + commission)? Don’t work out who works in department 10, department codes may change next week!

What has to be determined before the main results can be selected?

Carefully check the results of your answer against the original tables.

**SELECT E.ENAME, (E.SAL + NVL(E.COMM, 0)) AS INCOME FROM EMP E INNER JOIN DEPT D ON E.DEPTNO = D.DEPTNO WHERE D.LOC = 'NEW YORK' AND (E.SAL + NVL(E.COMM, 0)) = (SELECT MAX(E.SAL + NVL(E.COMM, 0)) AS INCOME FROM EMP E INNER JOIN DEPT D ON E.DEPTNO = D.DEPTNO WHERE D.LOC = 'NEW YORK');**

**G4** List the names of the people who work with Jones (you may use the fact that his Employee number is 7566 as there may be more than one 'JONES' on the list) in his department.

**SELECT** ENAME **FROM** EMP **WHERE** EMPNO != ‘7566’ **AND DEPTNO**

**=(SELECT** DEPTNO **FROM** EMP **WHERE EMPNO = ‘7566’**

Have you displayed 5 employees? JONES really work with himself?

**);**

Have you displayed JONES? If so, does

**G5** List the names of anyone who started work on the same date as FORD.

**SELECT \* FROM EMP WHERE HIREDATE = (SELECT HIREDATE FROM EMP WHERE EMPNO = '7902');**

SQL Workbook 65 September 2020

Section G

**G6** List all employees who have the same basic salary as FORD.

**SELECT \* FROM EMP WHERE SAL = (SELECT SAL FROM EMP WHERE EMPNO = '7902');**

**G7** What is the average salary of those employees on Grade 2? Use a subquery in your solution.

**SELECT AVG(SAL) FROM EMP WHERE ENAME IN (SELECT ENAME FROM EMP INNER JOIN SALGRADE ON SAL BETWEEN LOSAL AND HISAL WHERE GRADE = 2);**

Now produce a solution without using a subquery.

**SELECT AVG(SAL) FROM EMP RIGHT OUTER JOIN SALGRADE ON SAL BETWEEN LOSAL AND HISAL AND GRADE = 2;**

**G8** Find the name and salary of the people who earn more than the average salary of those on grade 2.

**SELECT ENAME, SAL FROM EMP WHERE SAL > (SELECT AVG(SAL) FROM EMP RIGHT OUTER JOIN SALGRADE ON SAL BETWEEN LOSAL AND HISAL AND GRADE = 2);**

SQL Workbook 66 September 2020