**RETRIEVING DATA FROM MULTIPLE TABLES**

Creating (inner) joins to ensure the correct data is displayed

Creating and using table aliases

**ESSENTIAL = 1-6,    VERY USEFUL = 7**

When more than one table is used in a SQL SELECT statement, if no join condition is specified to tell the database how to join the two tables together, every row from one table is matched to every row in the other table and this is called the Cartesian product.

SELECT         ename, dname

FROM            emp, dept

WHERE          ename LIKE 'S%'

ORDER BY    ename;

|  |  |
| --- | --- |
| **ENAME** | **DNAME** |
| SCOTT | ACCOUNTING |
| SCOTT | RESEARCH |
| SCOTT | OPERATIONS |
| SCOTT | SALES |
| SMITH | ACCOUNTING |
| SMITH | OPERATIONS |
| SMITH | SALES |
| SMITH | RESEARCH |

Here both the employees with names starting with S have been joined to each of the available departments which is clearly wrong.

To resolve this you must specify the matching column(s) in the two tables which is the primary key of the master table to the foreign key for the detail table.  Here, since deptno is the primary key of the parent table in the relationship (one dept has many employees):

1. Display the employee name for all employees beginning with S along with the name of the department they work for.

SELECT ename, dname

FROM emp

INNER JOIN dept ON emp.deptno = dept.deptno

WHERE ename LIKE 'S%'

ORDER BY ename;

|  |  |
| --- | --- |
| **ENAME** | **DNAME** |
| SCOTT | RESEARCH |
| SMITH | RESEARCH |

For every employee with an ename starting with S, find the deptno for that employee and then look up the record with the same deptno in the dept table returning the dname.

2.        Now add the deptno to the last query and use table aliases.

SELECT e.ename, d.deptno, d.dname

FROM emp e

INNER JOIN dept d ON e.deptno = d.deptno

WHERE e.ename LIKE 'S%'

ORDER BY e.ename;

|  |  |  |
| --- | --- | --- |
| **ENAME** | **DEPTNO** | **DNAME** |
| SCOTT | 20 | RESEARCH |
| SMITH | 20 | RESEARCH |

If **table aliases** are declared in the FROM clause (by typing an abbreviation immediately after the table name), these can be used throughout the statement in place of the table name.  Here e is the table alias for emp and d the table alias for dept.

Note: To improve readability, table aliases are used to prefix all column names in the statement.

3.        Display the customer name, order id, order date and item id for all orders placed between the 1st Jan 2005 and 12th Jan 2005. Order rows by customer name and then order id.

SELECT c.name, o.ordid, o.orderdate, i.itemid

FROM customer c

INNER JOIN ord o ON c.custid = o.custid

INNER JOIN item i ON o.ordid = i.ordid

WHERE o.orderdate BETWEEN '2005-01-01' AND '2005-01-12'

ORDER BY c.name, o.ordid;

|  |  |  |  |
| --- | --- | --- | --- |
| **NAME** | **ORDID** | **ORDERDATE** | **ITEMID** |
| TKB SPORT SHOP | 610 | 2005-01-07 | 1 |
| TKB SPORT SHOP | 610 | 2005-01-07 | 2 |
| TKB SPORT SHOP | 610 | 2005-01-07 | 3 |
| VOLLEYRITE | 611 | 2005-01-11 | 1 |

You can use as many tables as you like in a SELECT statement provided that they are all joined correctly using primay and foreign key relationships (see ERD).   Here customer is joined to ord and ord is joined to item.

4. Now add the product description to the last query.

SELECT c.name, o.ordid, o.orderdate, i.itemid, p.descrip

FROM customer c

INNER JOIN ord o ON c.custid = o.custid

INNER JOIN item i ON o.ordid = i.ordid

INNER JOIN product p ON p.prodid = i.prodid

WHERE o.orderdate BETWEEN '2005-01-01' AND '2005-01-12'

ORDER BY c.name, o.ordid;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NAME** | **ORDID** | **ORDERDATE** | **ITEMID** | **DESCRIP** |
| TKB SPORT SHOP | 610 | 2005-01-07 | 1 | ACE TENNIS RACKET I |
| TKB SPORT SHOP | 610 | 2005-01-07 | 2 | ACE TENNIS BALLS-3 PACK |
| TKB SPORT SHOP | 610 | 2005-01-07 | 3 | ACE TENNIS NET |
| VOLLEYRITE | 611 | 2005-01-11 | 1 | ACE TENNIS RACKET II |

5.         **Compound keys:**

  Display the order id, item id, item qty and delivered qty for order 612 and 614

SELECT i.ordid, i.itemid, i.qty, di.qty

FROM item i

INNER JOIN delivered\_item di ON i.ordid = di.ordid

AND i.itemid = di.itemid

WHERE i.ordid IN (612, 614);

|  |  |  |  |
| --- | --- | --- | --- |
| **ORDID** | **ITEMID** | **QTY** | **QTY** |
| 612 | 1 | 100 | 75 |
| 612 | 1 | 100 | 25 |
| 612 | 2 | 20 | 20 |
| 612 | 3 | 150 | 100 |
| 612 | 3 | 150 | 50 |
| 612 | 4 | 100 | 100 |
| 614 | 1 | 444 | 400 |
| 614 | 1 | 444 | 44 |
| 614 | 2 | 1000 | 1000 |
| 614 | 3 | 1000 | 1000 |

For joins between tables with a compound foreign key relationship it is essential that all elements of the key are included in the join.   The foreign key relationship between item and delivered\_item is ordid and itemid – so the join between these two tables has to match on both columns.

6.         Now we want to refine the last query to only show items that were delivered in more than one consignment (i.e. where the item qty does not equal the delivered qty.

SELECT i.ordid, i.itemid, i.qty, di.qty

FROM item i

INNER JOIN delivered\_item di ON i.ordid = di.ordid

AND i.itemid = di.itemid

WHERE i.qty <> di.qty

AND i.ordid IN (612, 614);

|  |  |  |  |
| --- | --- | --- | --- |
| **ORDID** | **ITEMID** | **QTY** | **QTY** |
| 612 | 1 | 100 | 75 |
| 612 | 1 | 100 | 25 |
| 612 | 3 | 150 | 100 |
| 612 | 3 | 150 | 50 |
| 614 | 1 | 444 | 400 |
| 614 | 1 | 444 | 44 |

7. Now we want to change the column headings for the two qty fields to make them clearer and add the product description and the delivery date to the above query.

SELECT i.ordid, i.itemid, p.descrip, i.qty AS 'Item Qty', di.qty AS 'Delivered Qty', d.deliverydate

FROM item i

INNER JOIN product p ON i.prodid = p.prodid

INNER JOIN delivered\_item di ON i.ordid = di.ordid

AND i.itemid = di.itemid

INNER JOIN delivery d ON d.delid = di.delid

WHERE i.qty <> di.qty

AND i.ordid IN (612, 614);

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ORDID** | **ITEMID** | **DESCRIP** | **ITEM QTY** | **DELIVERED QTY** | **DELIVERYDATE** |
| 612 | 1 | ACE TENNIS RACKET I | 100 | 75 | 2005-01-25 |
| 612 | 1 | ACE TENNIS RACKET I | 100 | 25 | 2005-02-10 |
| 612 | 3 | SP JUNIOR RACKET | 150 | 100 | 2005-01-25 |
| 612 | 3 | SP JUNIOR RACKET | 150 | 50 | 2005-02-10 |
| 614 | 1 | ACE TENNIS RACKET I | 444 | 400 | 2005-02-02 |
| 614 | 1 | ACE TENNIS RACKET I | 444 | 44 | 2005-02-14 |

**You should always refer to the ERD in order to determine how best to join tables.**