

Structured Query Language

SELECT statement in Oracle

SQL

- SQL stands for Structured Query Language.
- There is a standard SQL called the American National Standards Institute's 2003 Standard SQL (ANSI:2003 SQL)
- Most database vendors cover much of the standard, but do not adhere to it completely.
 - Note, when you start using Oracle SQL, you will see that many of the functions are different.

SELECT

- This statement is very versatile and is the single retrieval mechanism.
- Its basic components are
 - SELECT
 - Field-list
 - FROM
 - Table-list

SELECT field-list

- The field-list in a SELECT statement can be:
 - A wildcard character * to denote all eligible fields.
 - A column name that is unique to one of the tables in the table list.
 - A table-name.column-name to specify a column from a specific table from the table list.
 - A schema-name.table-name.column-name to specify a column in a specific table in a specific schema
 - Please note: Oracle calls the domain in which the tables reside a schema. SQL Server calls this domain a database.
 - A derived field

Derived fields in the SELECT field-list

- Fields can be derived by:
 - Performing calculations on column-fields from the table-list.
 - E.g. `unitprice * quantity` as `linecost`
 - Using functions on the column-fields from the table list.
 - E.g. `day(orderdate)`
 - Using database provided functions.
 - `Lower('Harry')` *will return* 'harry'.
 - `Months_between(sysdate, sysdate-31)` *will return* 1.

Oracle system / session variables

- Oracle uses Dual as a 'dummy' table for system variables, notably SYSDATE (*current date and time*) and USER(*user of current session*)).
- These variables can be selected from DUAL or used as stand-alone:
- Example uses:
 - Select sysdate from dual;
 - Select user from dual;
- Or
 - Select sysdate – supplierorderdate as 'Days working' from employees.

Table-list

- The number of tables involved in a query depends on:
 - The contents of the field-list
 - i.e. Any table that holds one of the fields, must be in the list.
 - Any filters involved
 - i.e. Any table that holds a field that is being used in a 'WHERE' condition must be in the list.
 - Any joining tables
 - If a traversal is required to get from one table to another, even if no field from the intermediate table is used, the table join details must be included.
 - For example, if I want a list of all job titles and department names for those jobs, I need to join through the employees table.

To pick out rows:

- This is called projection and is done using the WHERE clause.
- This clause goes after the basic select:
- `SELECT column-list FROM table-list WHERE condition`

Conditions

- The conditions can include:
- >,<,<>,<=,>=, !=, NOT, between, IS NULL, IS LIKE.
- The **IS NULL** returns a true if the value in the column is null, and a false otherwise.
- You will NEVER get anything if you use the condition WHERE column = NULL
 - Null means undefined. You cannot equate to undefined!

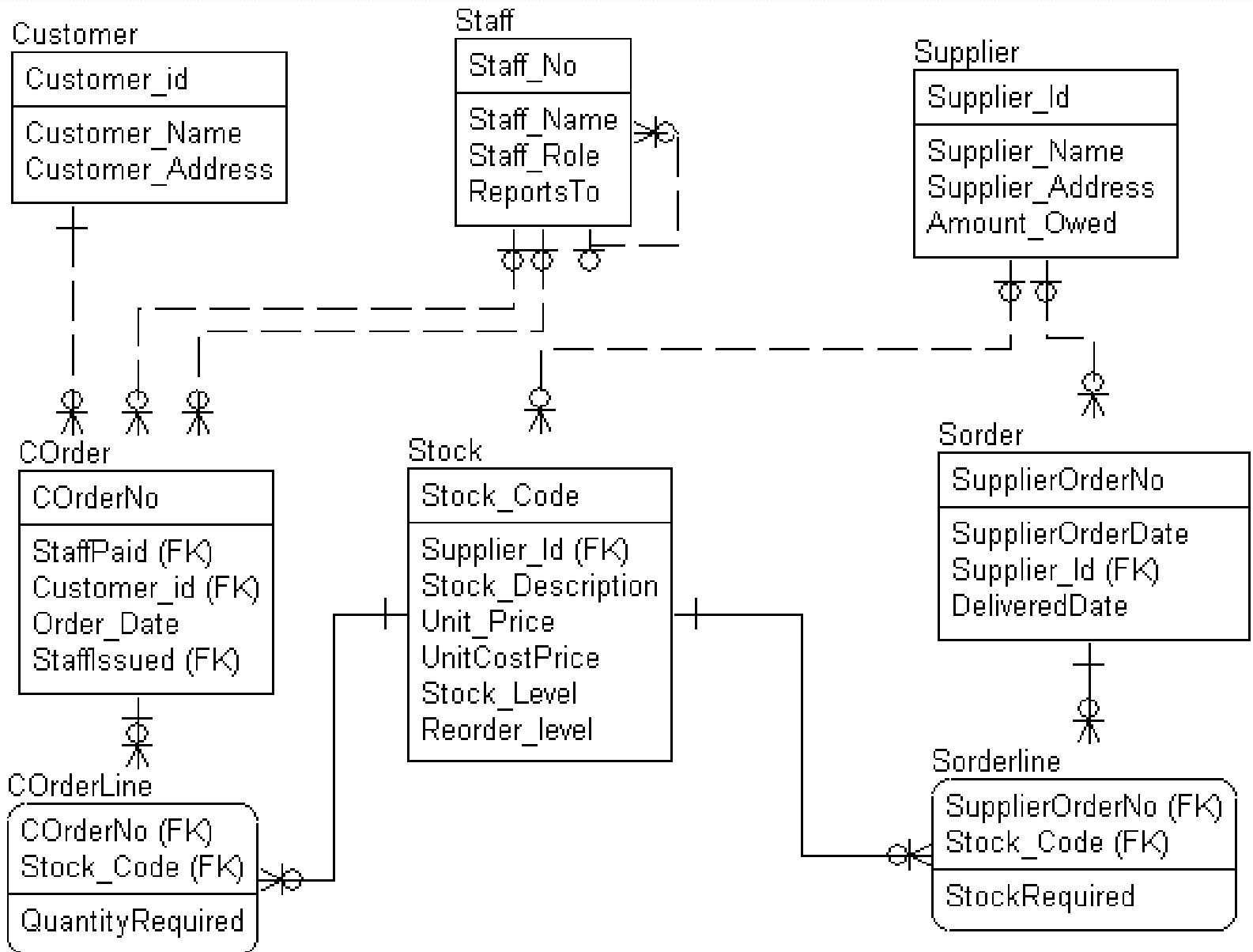
LIKE

- LIKE allows us to match patterns in strings.
- Wildcard characters can be used to represent
 - A character from a string _
 - A variable length substring from a string %.

Manipulating ROWS

- The WHERE clause allows to choose from specific ROWS in our query:
 - `SELECT stock_description from stock where stock_level BETWEEN 10 and 100;`
- Ordering your data
 - To sort the output by a particular column, add the suffix ORDER BY column-name
 - E.g. `SELECT stock_description from stock where stock_level BETWEEN 10 and 100 order by stock_level;`
- You can reverse the order:
 - E.g. `SELECT stock_description from stock where stock_level BETWEEN 10 and 100 order by stock_level desc;`

Builder schema tables



"
If you can't find it, go to Joe!"

The Docket

Joe's Builders' Supplies

(01) 8437083 Main St., Malmont, D27.

Docket No: 196

Stock code.	Quantity	Description	Unit Price	VAT (20%)	Total Price
A642	4	4"x4" treated timber	€9.50	€1.90	€45.60
J555	1	Box of 6" phillips screws (100)	€5.00	€1.00	€6.00
J501	1	Phillips head screwdriver	€8.00	€1.60	€9.60

Total Amount Due: €61.20

Docket paid signature Joe Bloggs (Staff)

Staff No: 124 Staff Name Joe Bloggs Staff Role: Yard foreman

Items received signature Andy Handy (Customer)

DRAFT

Some fields that were eliminated

- Single row aggregates:
 - VAT@20%
 - Total Price
- Multiple Row aggregates:
 - TotalAmountDue
- Other possible multi-row aggregates:
 - Total number of items bought.
 - Total number of types of items bought.

Non-base fields

- Derived fields
- Sum
- Average
- Count
- Summing group fields

Summing within a row

- To derive data within a row, just do the calculation as another field in the 'select':
 - $(\text{Stock.unit_price} * .2)$ as VatAt20
- To derive the total cost of one item of stock, we multiply it by 1.2 (i.e. including the VAT).

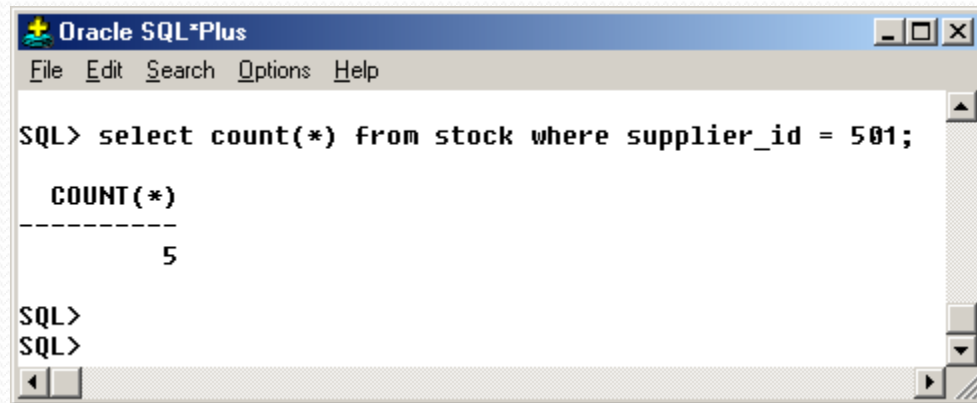
To aggregate over > 1 row

- Count rows
 - How many order lines are on that order?
 - How many times was that stock item ordered?
- Sum over rows
 - How much of that stock item has been sold?
 - i.e. sum the quantities in all the order lines matching the stock item.

Count the rows

- The count function counts rows:
 - `Select count(*) from stock;`
- This counts the number of rows in stock.
- I could also count the number of items of stock that were supplied by a single supplier:
 - `Select count(*) from stock where supplier_id = 501;`

Outcome



A screenshot of the Oracle SQL*Plus command-line interface. The window has a title bar that reads "Oracle SQL*Plus" and a menu bar with "File", "Edit", "Search", "Options", and "Help". The main text area shows a SQL query being executed: "SQL> select count(*) from stock where supplier_id = 501;". Below the query, the result is displayed as a table with one column, "COUNT(*)", and one row containing the value "5". The prompt "SQL>" appears twice at the bottom of the window, indicating the end of the query execution and the start of a new command line.

```
Oracle SQL*Plus
File Edit Search Options Help

SQL> select count(*) from stock where supplier_id = 501;

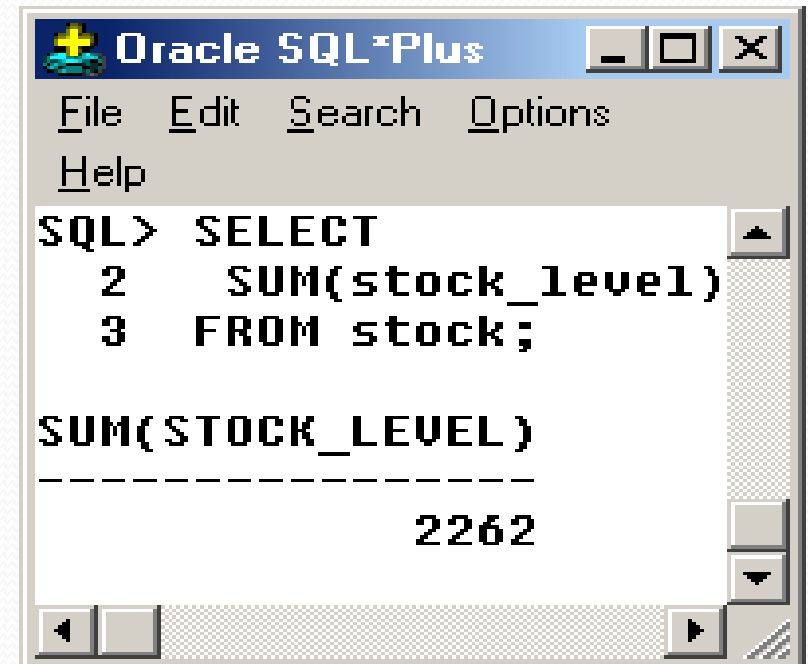
  COUNT(*)
-----
         5

SQL>
SQL>
```

Likewise, I can sum things

- How many physical items of stock have I got?

```
SELECT
    SUM(stock_level)
FROM stock;
```



The screenshot shows the Oracle SQL*Plus interface. The title bar reads "Oracle SQL*Plus". The menu bar includes "File", "Edit", "Search", "Options", and "Help". The command prompt shows the following SQL query being executed:

```
SQL> SELECT
      2    SUM(stock_level)
      3  FROM stock;
```

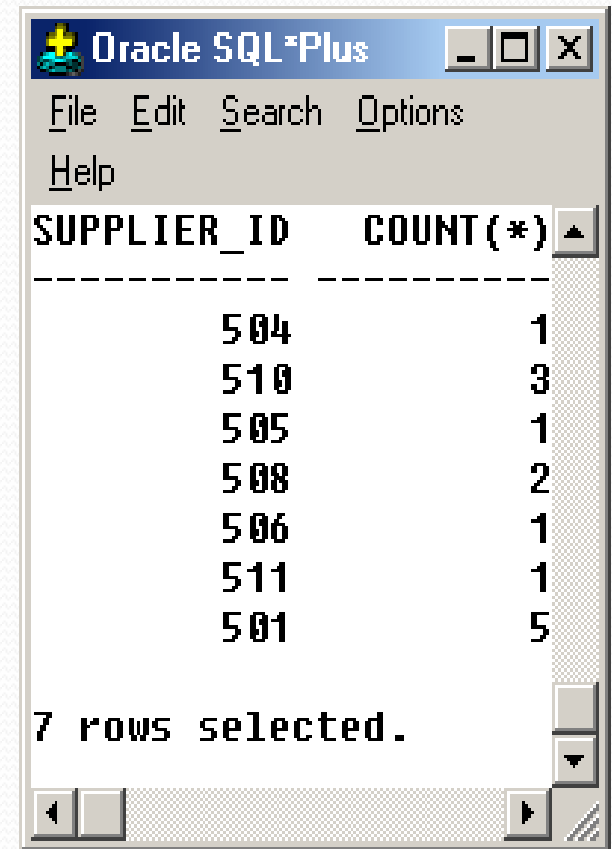
The output of the query is displayed below the prompt:

```
SUM(STOCK_LEVEL)
-----
                2262
```

Grouping

- Sometimes we need to have sub-groups. This is done by using the 'GROUP BY' clause.
- We can count all stock types for each supplier:

```
SELECT supplier_id,  
       COUNT (*)  
FROM stock  
GROUP BY supplier_id;
```



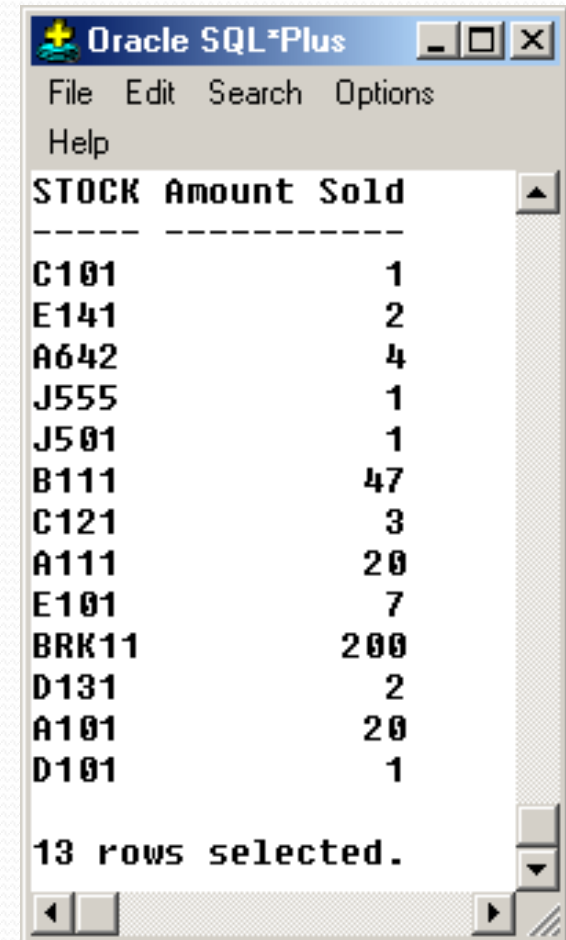
The screenshot shows the Oracle SQL*Plus interface. The title bar reads 'Oracle SQL*Plus'. The menu bar includes 'File', 'Edit', 'Search', 'Options', and 'Help'. The query results are displayed in a table with two columns: 'SUPPLIER_ID' and 'COUNT (*)'. The data rows are as follows:

SUPPLIER_ID	COUNT (*)
504	1
510	3
505	1
508	2
506	1
511	1
501	5

At the bottom of the window, it states '7 rows selected.'.

Summing over lines

```
SELECT
    stock_code,
    sum(quantityrequired)
    AS "Amount Sold"
FROM
    corderline
GROUP BY stock_code;
```



The screenshot shows the Oracle SQL*Plus interface. The title bar reads "Oracle SQL*Plus". The menu bar includes "File", "Edit", "Search", "Options", and "Help". The main display area shows the results of the query in a table format with columns "STOCK" and "Amount Sold". There are 13 rows of data. At the bottom, it states "13 rows selected.".

STOCK	Amount Sold
C101	1
E141	2
A642	4
J555	1
J501	1
B111	47
C121	3
A111	20
E101	7
BRK11	200
D131	2
A101	20
D101	1

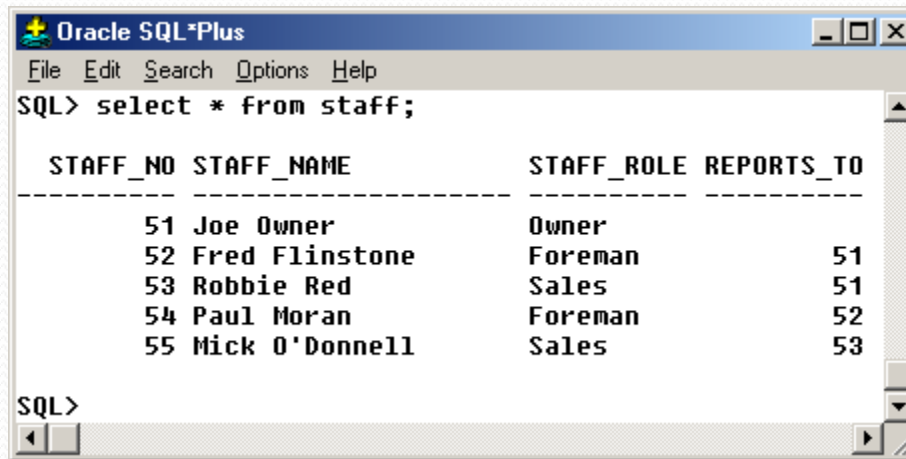
How do I?

- Make decisions based on group conditions?
 - E.g To find
 - What is the average number of corderlines in a corder?
 - Which corders have more than 2 corderlines?
- Select corderno, count(*) from corderline
group by corderno
having count(*) >2

Some relationship types

Using the Builder2 schema

The staff table.

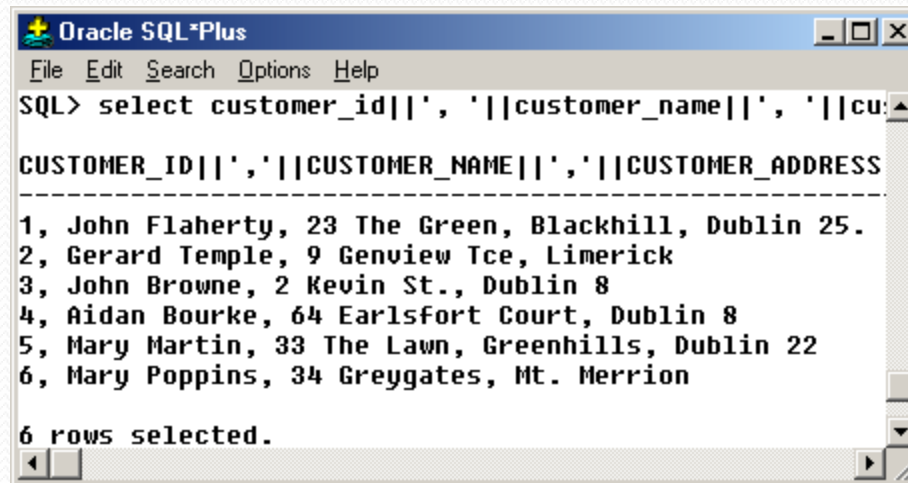


A screenshot of the Oracle SQL*Plus command-line interface. The window title is "Oracle SQL*Plus". The menu bar includes "File", "Edit", "Search", "Options", and "Help". The command prompt shows the command "SQL> select * from staff;". The output is a table with four columns: STAFF_NO, STAFF_NAME, STAFF_ROLE, and REPORTS_TO. The data rows are as follows:

STAFF_NO	STAFF_NAME	STAFF_ROLE	REPORTS_TO
51	Joe	Owner	
52	Fred Flinstone	Foreman	51
53	Robbie Red	Sales	51
54	Paul Moran	Foreman	52
55	Mick O'Donnell	Sales	53

The command prompt at the bottom shows "SQL>" and a scroll bar is visible on the right side of the window.

The customer table

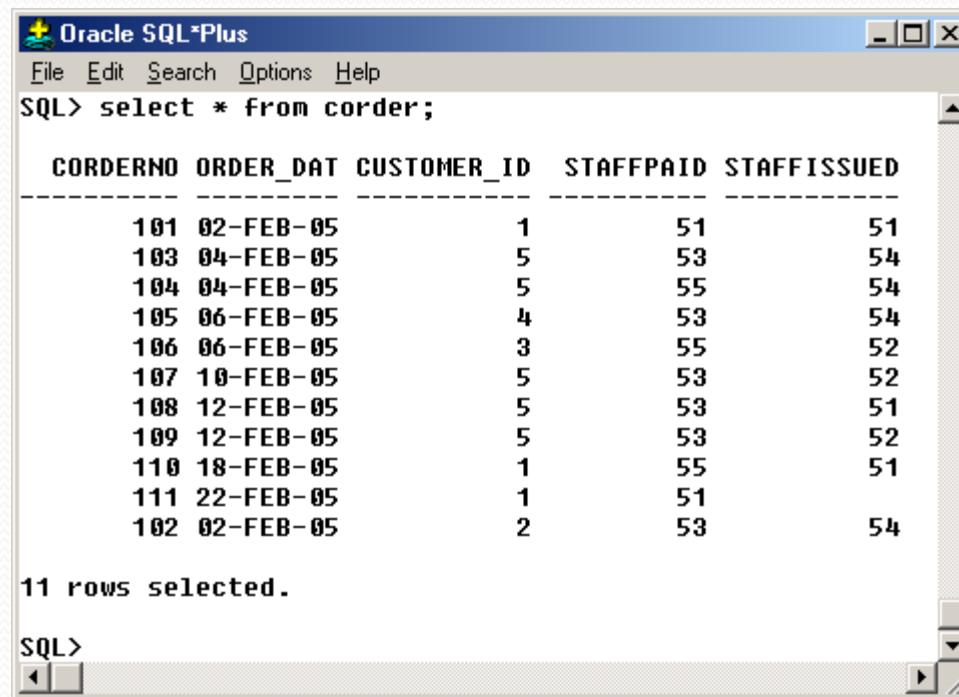


The screenshot shows a window titled "Oracle SQL*Plus" with a menu bar (File, Edit, Search, Options, Help). The command prompt shows the following SQL query and its results:

```
SQL> select customer_id||', '||customer_name||', '||cu:
CUSTOMER_ID||', '||CUSTOMER_NAME||', '||CUSTOMER_ADDRESS
-----
1, John Flaherty, 23 The Green, Blackhill, Dublin 25.
2, Gerard Temple, 9 Genview Tce, Limerick
3, John Browne, 2 Kevin St., Dublin 8
4, Aidan Bourke, 64 Earlsfort Court, Dublin 8
5, Mary Martin, 33 The Lawn, Greenhills, Dublin 22
6, Mary Poppins, 34 Greygates, Mt. Merrion

6 rows selected.
```

The customer order (corder) table



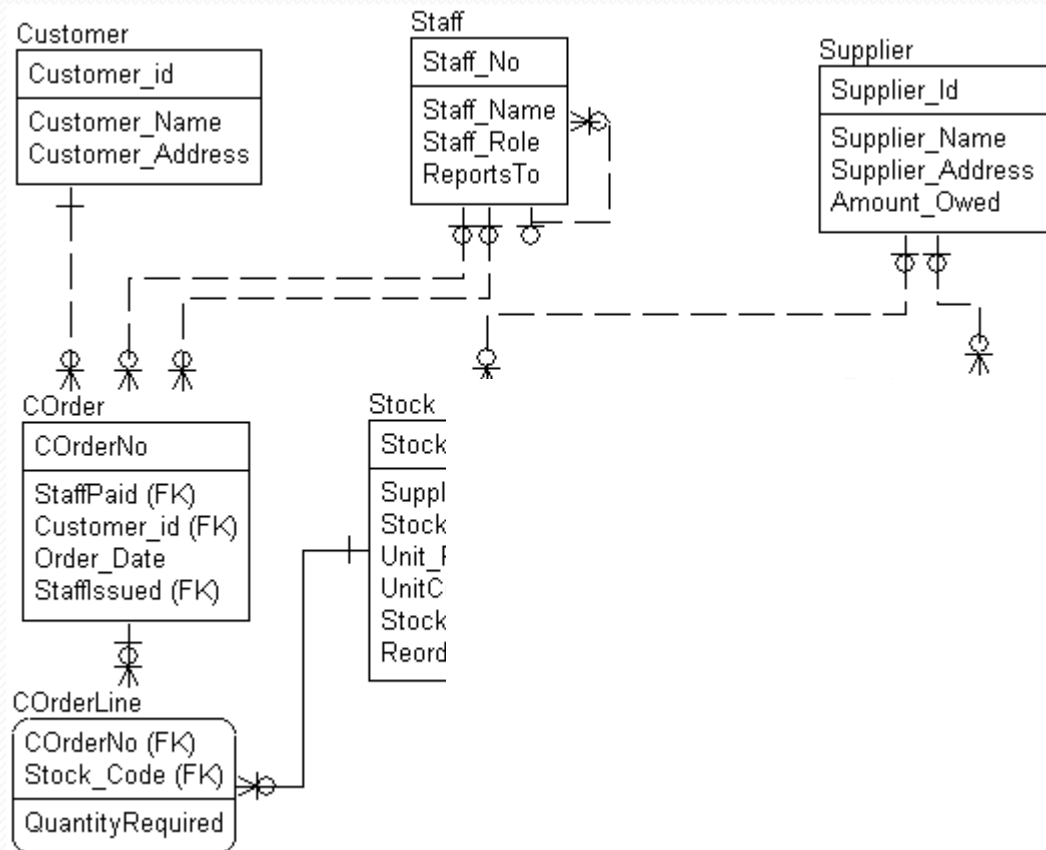
A screenshot of the Oracle SQL*Plus command-line interface. The window title is "Oracle SQL*Plus". The menu bar includes "File", "Edit", "Search", "Options", and "Help". The command prompt shows the command "SQL> select * from corder;". Below the command, the results of the query are displayed in a table format. The table has five columns: CORDERNO, ORDER_DAT, CUSTOMER_ID, STAFFPAID, and STAFFISSUED. There are 11 rows of data. At the bottom of the window, it says "11 rows selected." and the prompt "SQL>" is visible.

CORDERNO	ORDER_DAT	CUSTOMER_ID	STAFFPAID	STAFFISSUED
101	02-FEB-05	1	51	51
103	04-FEB-05	5	53	54
104	04-FEB-05	5	55	54
105	06-FEB-05	4	53	54
106	06-FEB-05	3	55	52
107	10-FEB-05	5	53	52
108	12-FEB-05	5	53	51
109	12-FEB-05	5	53	52
110	18-FEB-05	1	55	51
111	22-FEB-05	1	51	
102	02-FEB-05	2	53	54

11 rows selected.

SQL>

Now let's join them...



What sort of joins?

- One to many
- 2:many
- Many:Many
- Zero or one to many
- Reflexive

How do I write...?

- List all staff who took payment for orders and the orders for which they took payment.
- List all staff who were involved with orders and the orders they were involved with.
- List all staff and the customers they dealt with.
- List each staff member's name, along with the name of his / her boss.

Creating Views

- The following are the fields in the stock table:
- Which of them do you think the customer needs to see?
- Is the Supplier_id enough information for the customer?
- Design a customer view.

Stock

Stock_Code
Supplier_Id (FK)
Stock_Description
Unit_Price
UnitCostPrice
Stock_Level
Reorder_level

Inner and Outer Joins

- An inner join occurs when records are selected from two tables and the values in one column from the first table are also found in a similar column in the second table.

Oracle Inner Joins

- The tables to be joined are listed in the FROM clause and then related together in the WHERE clause:

```
SELECT staff_name, corderno
FROM staff, corder
WHERE staff_no = corder.staffpaid;
```

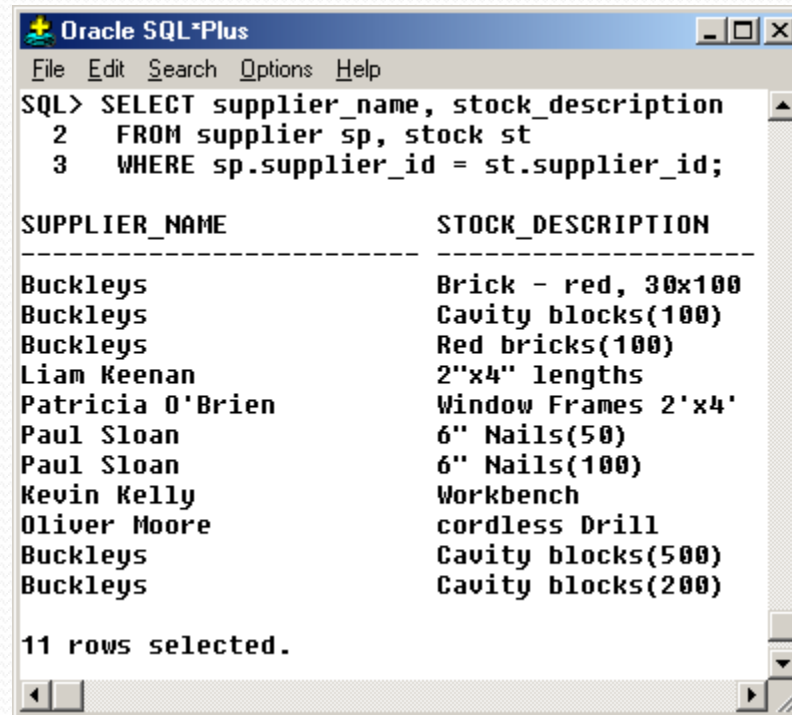
- Or

```
SELECT supplier_name, stock_description
FROM supplier, stock
WHERE supplier.supplier_id = stock.supplier_id;
```

- Or

```
SELECT supplier_name, stock_description
FROM supplier sp, stock st
WHERE sp.supplier_id = st.supplier_id;
```

Outcome...



The screenshot shows the Oracle SQL*Plus application window. The title bar reads "Oracle SQL*Plus". The menu bar includes "File", "Edit", "Search", "Options", and "Help". The command prompt shows the following SQL query:

```
SQL> SELECT supplier_name, stock_description
2   FROM supplier sp, stock st
3   WHERE sp.supplier_id = st.supplier_id;
```

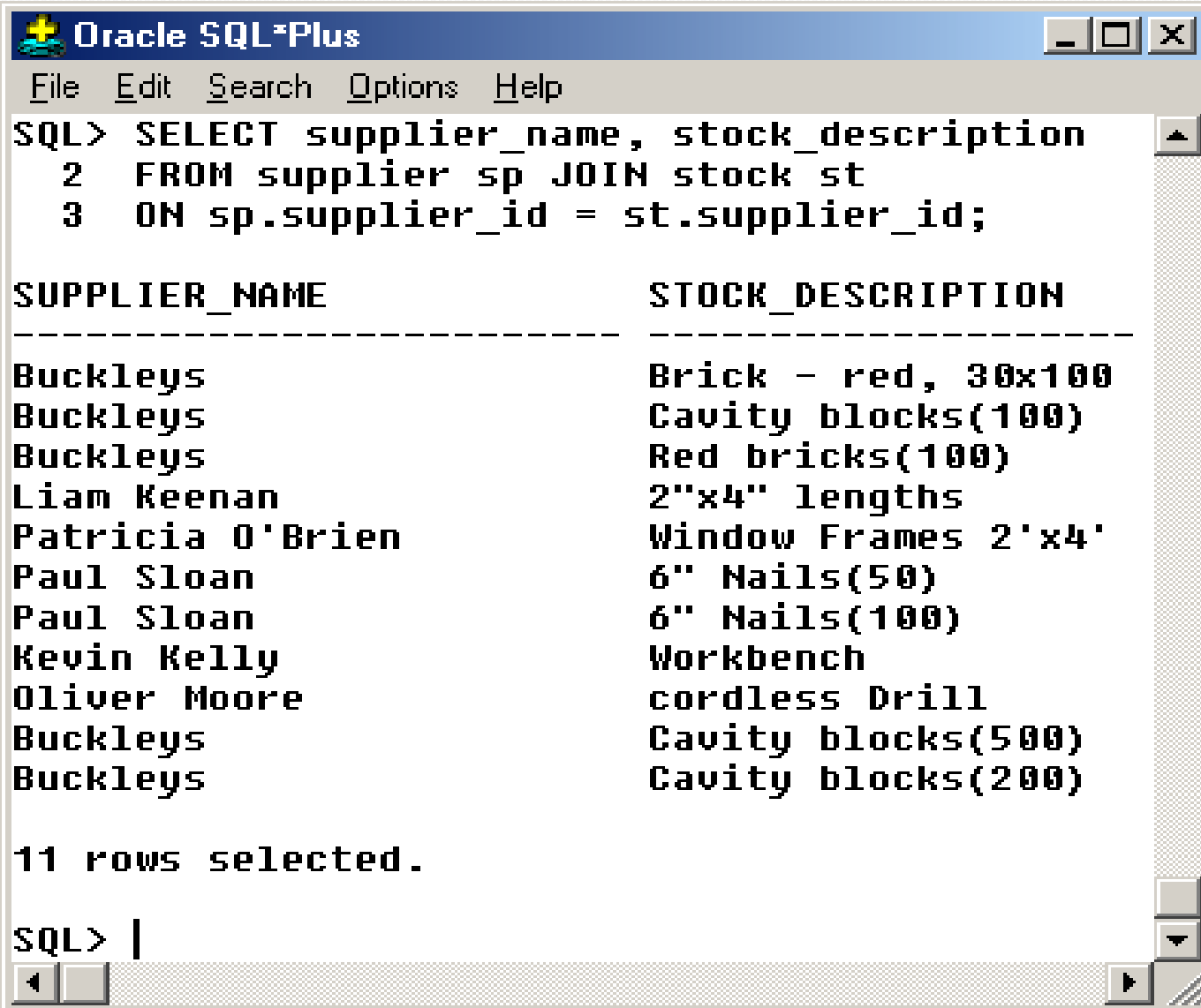
The query result is displayed in a table with two columns: "SUPPLIER_NAME" and "STOCK_DESCRIPTION". The table contains 11 rows of data. At the bottom of the window, it states "11 rows selected."

SUPPLIER_NAME	STOCK_DESCRIPTION
Buckleys	Brick - red, 30x100
Buckleys	Cavity blocks(100)
Buckleys	Red bricks(100)
Liam Keenan	2"x4" lengths
Patricia O'Brien	Window Frames 2'x4'
Paul Sloan	6" Nails(50)
Paul Sloan	6" Nails(100)
Kevin Kelly	Workbench
Oliver Moore	cordless Drill
Buckleys	Cavity blocks(500)
Buckleys	Cavity blocks(200)

ANSI Inner joins

- American National Standards Institute (ANSI) has a different join:
- A simple join can be specified with an ON or a USING statement. The columns to be joined on will be listed.

Usir



The screenshot shows the Oracle SQL*Plus application window. The title bar reads "Oracle SQL*Plus". The menu bar includes "File", "Edit", "Search", "Options", and "Help". The command window contains the following SQL query:

```
SQL> SELECT supplier_name, stock_description
2 FROM supplier sp JOIN stock st
3 ON sp.supplier_id = st.supplier_id;
```

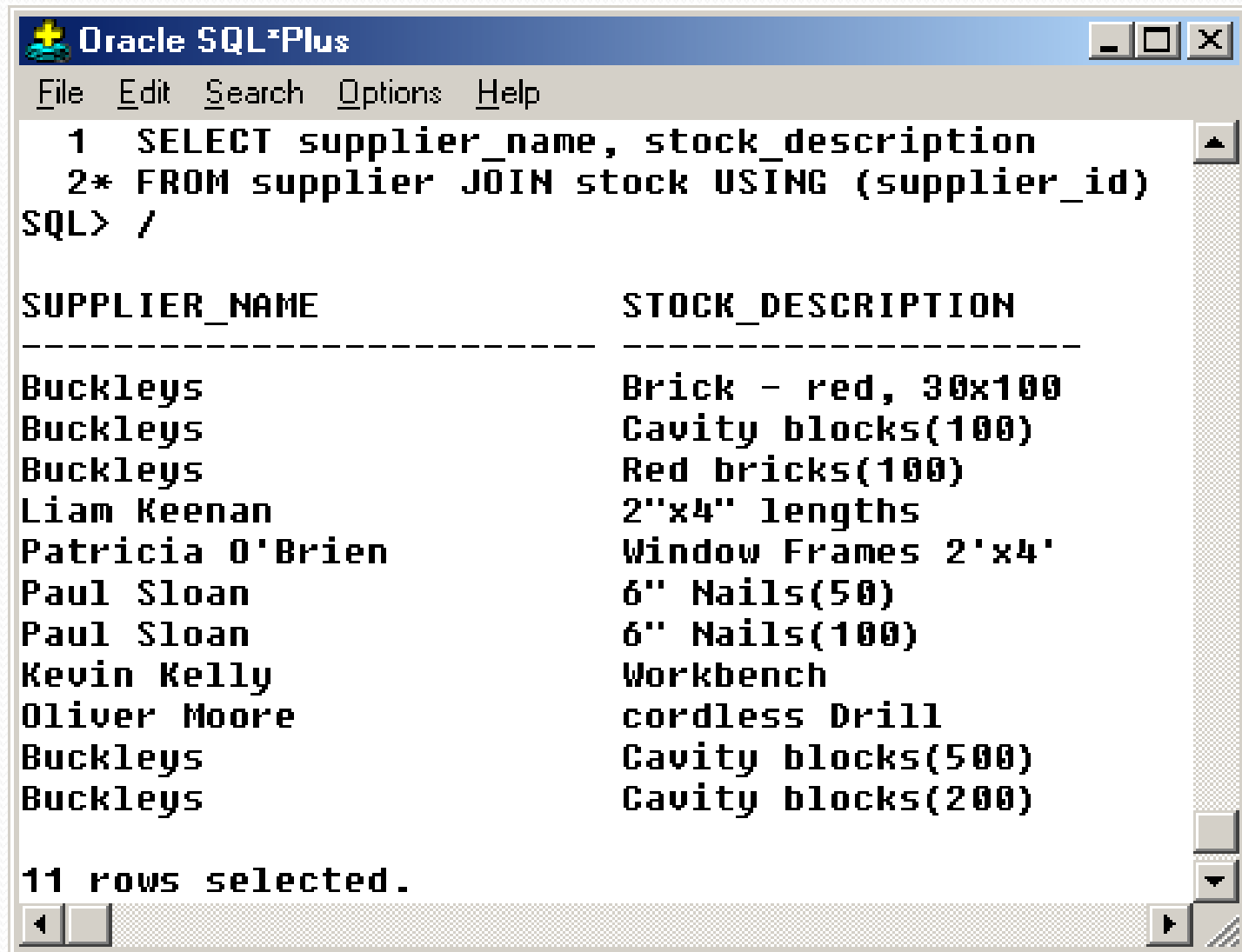
The results are displayed in a table with two columns: "SUPPLIER_NAME" and "STOCK_DESCRIPTION". There are 11 rows of data. Below the table, it states "11 rows selected." and the prompt "SQL> |" is visible.

SUPPLIER_NAME	STOCK_DESCRIPTION
Buckleys	Brick - red, 30x100
Buckleys	Cavity blocks(100)
Buckleys	Red bricks(100)
Liam Keenan	2"x4" lengths
Patricia O'Brien	Window Frames 2'x4'
Paul Sloan	6" Nails(50)
Paul Sloan	6" Nails(100)
Kevin Kelly	Workbench
Oliver Moore	cordless Drill
Buckleys	Cavity blocks(500)
Buckleys	Cavity blocks(200)

11 rows selected.

SQL> |

Using USING



The screenshot shows an Oracle SQL*Plus window with a menu bar (File, Edit, Search, Options, Help) and a command area. The command entered is a SQL query using the USING keyword for a join. The results are displayed in a table with two columns: SUPPLIER_NAME and STOCK_DESCRIPTION. There are 11 rows of data. At the bottom, it states '11 rows selected.'.

```
Oracle SQL*Plus
File Edit Search Options Help
1 SELECT supplier_name, stock_description
2* FROM supplier JOIN stock USING (supplier_id)
SQL> /
```

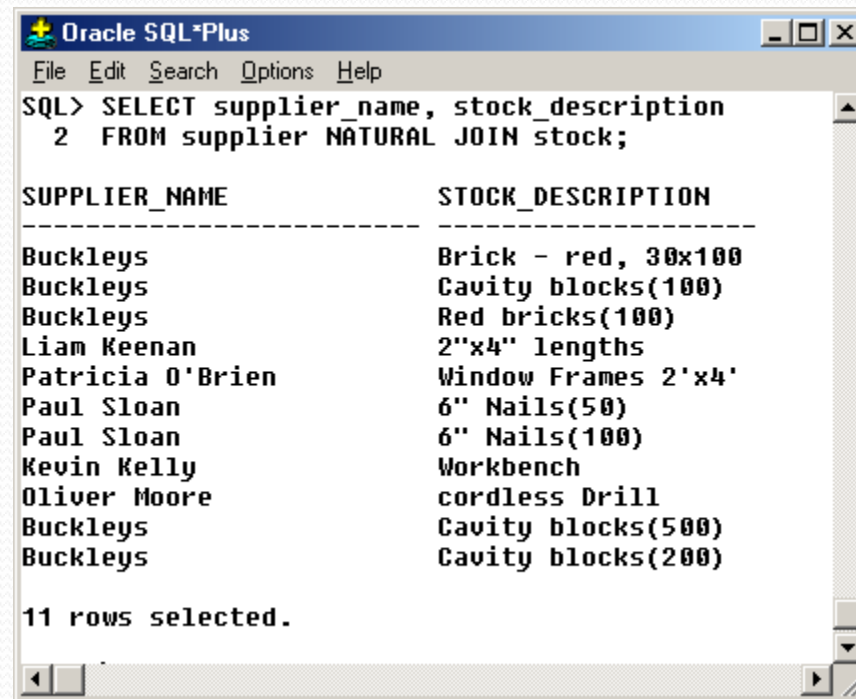
SUPPLIER_NAME	STOCK_DESCRIPTION
Buckleys	Brick - red, 30x100
Buckleys	Cavity blocks(100)
Buckleys	Red bricks(100)
Liam Keenan	2"x4" lengths
Patricia O'Brien	Window Frames 2'x4'
Paul Sloan	6" Nails(50)
Paul Sloan	6" Nails(100)
Kevin Kelly	Workbench
Oliver Moore	cordless Drill
Buckleys	Cavity blocks(500)
Buckleys	Cavity blocks(200)

11 rows selected.

ANSI Natural join

- The columns to be joined are not specified, but rather are resolved by Oracle. They must be similarly named in the tables to be joined.
- However, this is not recommended, as sometimes the same name can mean different things in different contexts.

Outcome...



A screenshot of the Oracle SQL*Plus application window. The title bar reads "Oracle SQL*Plus". The menu bar includes "File", "Edit", "Search", "Options", and "Help". The command prompt shows the SQL query: "SQL> SELECT supplier_name, stock_description 2 FROM supplier NATURAL JOIN stock;". The result is displayed in a two-column table with headers "SUPPLIER_NAME" and "STOCK_DESCRIPTION". There are 11 rows of data. At the bottom, it says "11 rows selected.".

SUPPLIER_NAME	STOCK_DESCRIPTION
Buckleys	Brick - red, 30x100
Buckleys	Cavity blocks(100)
Buckleys	Red bricks(100)
Liam Keenan	2"x4" lengths
Patricia O'Brien	Window Frames 2'x4'
Paul Sloan	6" Nails(50)
Paul Sloan	6" Nails(100)
Kevin Kelly	Workbench
Oliver Moore	cordless Drill
Buckleys	Cavity blocks(500)
Buckleys	Cavity blocks(200)

11 rows selected.



Oracle SQL*Plus



File Edit Search Options Help

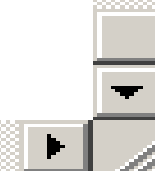
```
SQL> select corderno, supplierorderno FROM  
2 corderline NATURAL JOIN sorderline;
```

CORDERNO	SUPPLIERORDERNO
----------	-----------------

101	701
103	704
103	703
103	703
105	706
106	701

6 rows selected.

SQL>



Outer Joins

- An outer join can also return results from one table where the corresponding table did not have a matching value.
- For example, some suppliers are not supplying any stock.
- An outer join can show supplier details even if there is no corresponding stock.
 - See next slide.

```
Oracle SQL*Plus
File Edit Search Options Help
SQL> select supplier_name, stock_description
2 FROM supplier sp, stock st
3 WHERE sp.supplier_id = st.supplier_id(+);
```

SUPPLIER_NAME	STOCK_DESCRIPTION
Buckleys	Cavity blocks(100)
Buckleys	Red bricks(100)
Buckleys	Cavity blocks(200)
Buckleys	Brick - red, 30x100
Buckleys	Cavity blocks(500)
Brendan Moore	
James McGovern	
Liam Keenan	2"x4" lengths
Mary O'Brien	
Oliver Moore	cordless Drill
Robert O'Mahony	
Patricia O'Brien	Window Frames 2'x4'
June Browne	
Paul Sloan	6" Nails(100)
Paul Sloan	6" Nails(50)
Kevin Kelly	Workbench

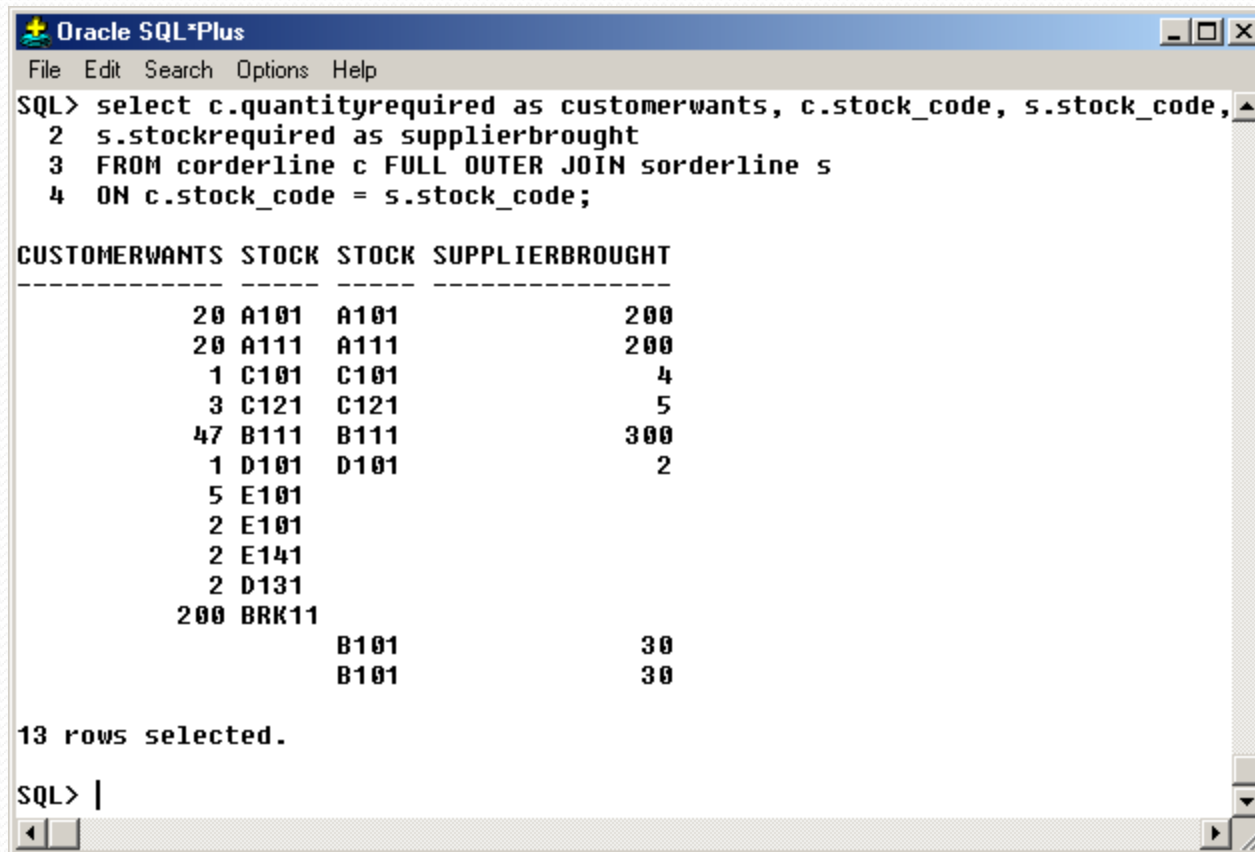
```
16 rows selected.

SQL>
```

ANSI outer join

- The ANSI outer join can be:
 - A left join (showing the full contents of the left table and any corresponding rows from the righthand table)
 - A right join (showing the full contents of the righthand table and any corresponding rows from the left table)
 - A full outer join (showing the full contents of the righthand and left hand tables, showing matches where appropriate).

Full outer join



The screenshot shows an Oracle SQL*Plus window with a menu bar (File, Edit, Search, Options, Help) and a command prompt. The user has entered a SQL query to perform a full outer join between two tables, CORDERLINE and SORDERLINE, on the stock_code column. The results are displayed in a table with four columns: CUSTOMERWANTS, STOCK, STOCK, and SUPPLIERBROUGHT. The results show 13 rows, including data from both tables and rows where the stock codes do not match.

```
SQL> select c.quantityrequired as customerwants, c.stock_code, s.stock_code,
2 s.stockrequired as supplierbrought
3 FROM corderline c FULL OUTER JOIN sorderline s
4 ON c.stock_code = s.stock_code;
```

CUSTOMERWANTS	STOCK	STOCK	SUPPLIERBROUGHT
20	A101	A101	200
20	A111	A111	200
1	C101	C101	4
3	C121	C121	5
47	B111	B111	300
1	D101	D101	2
5	E101		
2	E101		
2	E141		
2	D131		
200	BRK11		
		B101	30
		B101	30

13 rows selected.

```
SQL> |
```

To do difference

- Which stock items have never been ordered?
 - Either:
 - Use an Oracle outer join, between the stock and orderline tables, choosing null orderlines or
 - Use an ANSI standard outer join, between the stock and orderline tables, choosing null orderlines or
 - Use a sub-query, with the NOT IN clause or
 - Use Oracle MINUS.

This is the SQL...

-- ANSI outer join:

```
SELECT stock_code, stock_description  
FROM stock  
LEFT JOIN corderline USING (stock_code)  
WHERE corderno IS NULL;
```

-- ORACLE outer join

```
SELECT stock_code, stock_description  
FROM corderline c, stock s  
WHERE c.stock_code(+) = s.stock_code  
AND corderno IS NULL;
```

Look at the outer join first...

STOCK	STOCK_DESCRIPTION	CORDERNO
-----	-----	-----
A101	Cavity blocks (100)	101
A111	Red bricks (100)	106
A111	Red bricks (100)	102
A642	4"x4" treated timber	196
B101	2"x4" lengths	
B111	Window Frames 2'x4'	103
BRK11	Brick - red, 30x100	101
C101	6" Nails (50)	103
C121	6" Nails (100)	103
D101	Workbench	102
D101	Workbench	101

...continuing.

```
SELECT
stock_code,
stock_description,
corderno
FROM stock
LEFT JOIN
corderline USING
(stock_code);
```

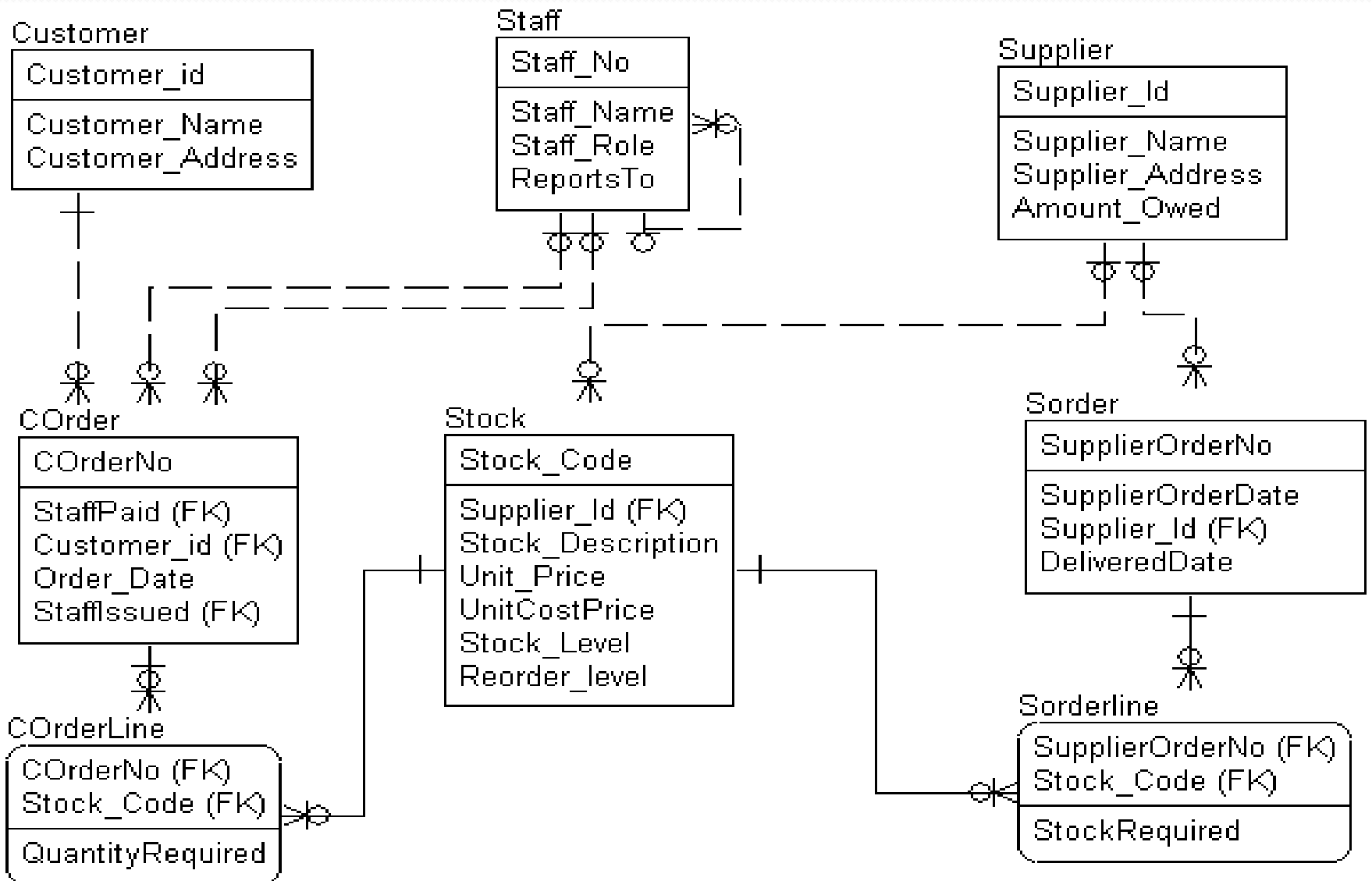
This is the SQL...

-- NOT IN

```
SELECT stock_code, stock_description  
FROM stock  
WHERE stock_code NOT IN  
      (SELECT stock_code FROM corderline);
```

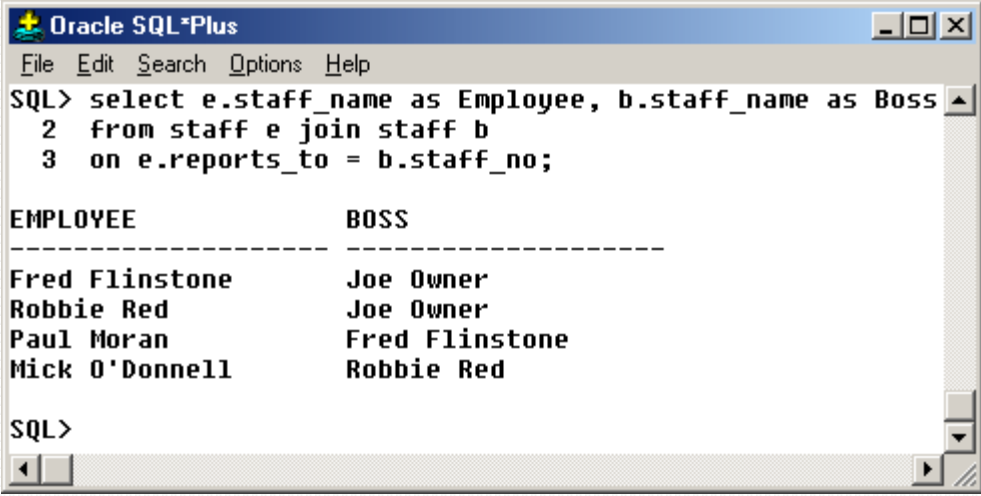
-- MINUS

```
SELECT stock_code FROM stock  
MINUS  
(SELECT stock_code FROM corderline);
```

Self Joins

Staff member to boss



The screenshot shows a window titled "Oracle SQL*Plus" with a menu bar (File, Edit, Search, Options, Help). The command prompt shows the following SQL query:

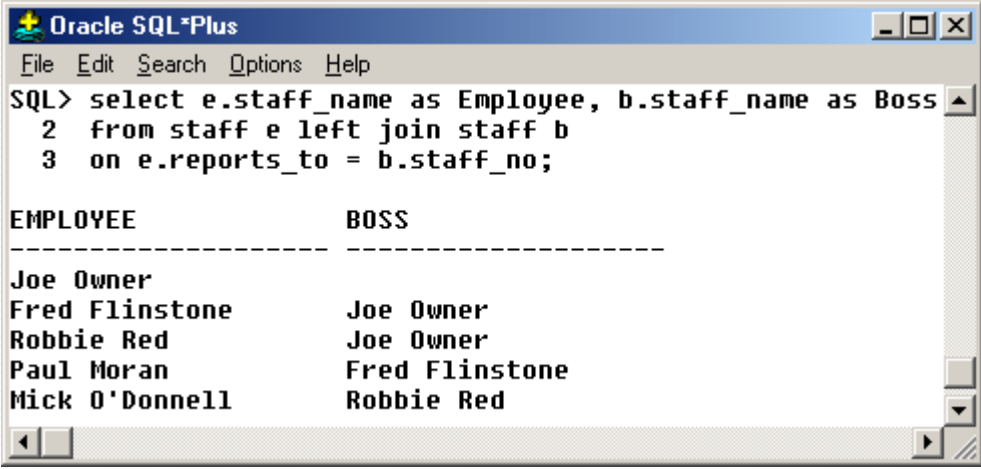
```
SQL> select e.staff_name as Employee, b.staff_name as Boss
2  from staff e join staff b
3  on e.reports_to = b.staff_no;
```

The results are displayed in two columns: EMPLOYEE and BOSS, separated by a dashed line.

EMPLOYEE	BOSS
Fred Flinstone	Joe Owner
Robbie Red	Joe Owner
Paul Moran	Fred Flinstone
Mick O'Donnell	Robbie Red

The prompt "SQL>" is visible at the bottom of the window.

All staff members + bosses



The screenshot shows an Oracle SQL*Plus window with a menu bar (File, Edit, Search, Options, Help) and a command area. The command area contains the following SQL query:

```
SQL> select e.staff_name as Employee, b.staff_name as Boss
2   from staff e left join staff b
3   on e.reports_to = b.staff_no;
```

Below the command area, the results of the query are displayed in a table format with two columns: EMPLOYEE and BOSS. The results are as follows:

EMPLOYEE	BOSS
Joe Owner	
Fred Flinstone	Joe Owner
Robbie Red	Joe Owner
Paul Moran	Fred Flinstone
Mick O'Donnell	Robbie Red

Exercises – task based queries

- Find out which suppliers have not delivered our orders (Give names).
- Find out which suppliers have never supplied any stock
 - I.e. we don't stock any items supplied by them (Give names).
- List the stock code and description of the most profitable item we stock (Give stock_code and description).
 - How are you making your judgement?
- Based on the last year's sales, which stock item are we selling most of (Give stock code and description)?
 - Write a query to display the quantity of the stock item, the quantity sold in the last month and the reorder level.