/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\* \*\*

\*\* CHAPTER 4. Practical 1. \*\*

\*\* JOINing related tables. INNER, Cross, Composite JOINs, Ambiguity, Aliases \*\*

\*\* \*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Use database0

SELECT \* FROM dept

SELECT \* FROM salesperson

SELECT \* FROM sale

SELECT \* FROM company

SELECT \* FROM contact

--p1) Display order number, order value and the NAME of the company each sale was made to.

-- If you are not sure what to JOIN on then run the 'sale' and 'company' SELECT statements

-- above and choose a sale at random and decide the name of the company it was made to and

-- write the code to mimic what you just did.

-- Use TABLE ALIASES, please, a good habit to get into early on.

-- Sort the answer set by company name. (8 rows)

-- Notice all 4 companies have been sold to (relevant later in the course).

SELECT s.order\_no, s.order\_value, c.name

FROM sale s

JOIN company c on c.company\_no = s.company\_no

ORDER BY c.name

order\_no order\_value name

----------- ----------- --------------------

100 7 Happy Heaters PLC

400 5 Happy Heaters PLC

300 12 Icicle Igloos Inc

700 3 Icicle Igloos Inc

200 6 Judo Jeans PLC

600 27 Judo Jeans PLC

800 3 Judo Jeans PLC

500 2 Kipper Kickers Inc

--p2) Totally new query.

-- Show all sales.

-- Display from each sale the order\_value and description columns.

-- Then, additionally, on the LEFT HAND SIDE of the display show the full name and dept number

-- of the person who made the sale. Use TABLE ALIASES please.

-- Notice the 8 sales have been made by 2 depts (3 people) only, relevant later.

-- (8 row result set)

SELECT sp.fname, sp.lname, sp.dept\_no, s.order\_value, s.description

FROM sale s

JOIN salesperson sp on sp.emp\_no = s.emp\_no

fname lname dept\_no order\_value description

--------------- --------------- ----------- ----------- --------------------------------------------------

Fred Goalie 3 7 Toshiba 6700 Pro

Fred Goalie 3 6 MS Office Professional \* 30

Fred Goalie 3 12 ScanPRO 4800 Scanner

Alan Brick 1 5 Modems and Cables etc

Fred Goalie 3 2 Laser printer

Ernest Flipper 3 27 Complete Desktop Publishing System

Alan Brick 1 3 SQL Server 2007 20 user licence

Fred Goalie 3 3 Printer cartridges

-- Now Copy/Paste your code in and

-- ADD the emp\_no of the person who made the sale as a FIRST column

-- and also sort the sales by this emp\_no.

SELECT sp.emp\_no, sp.fname, sp.lname, sp.dept\_no, s.order\_value, s.description

FROM sale s

JOIN salesperson sp on sp.emp\_no = s.emp\_no

order by sp.emp\_no

emp\_no fname lname dept\_no order\_value description

----------- --------------- --------------- ----------- ----------- --------------------------------------------------

10 Alan Brick 1 5 Modems and Cables etc

10 Alan Brick 1 3 SQL Server 2007 20 user licence

50 Ernest Flipper 3 27 Complete Desktop Publishing System

60 Fred Goalie 3 3 Printer cartridges

60 Fred Goalie 3 2 Laser printer

60 Fred Goalie 3 7 Toshiba 6700 Pro

60 Fred Goalie 3 6 MS Office Professional \* 30

60 Fred Goalie 3 12 ScanPRO 4800 Scanner

--p3) Copy/Paste your code from above).

-- Make 2 changes to your code:

-- Firstly, add the MANAGER of the seller of the sale as an extra FIRST column.

-- It is now broken, so get this working.

-- Secondly, restrict the answer set so that it lists only those sales

-- that contain the text 'printer' in their description. (2 rows)

--3.1)

SELECT d.manager, sp.emp\_no, sp.fname, sp.lname, sp.dept\_no, s.order\_value, s.description

FROM sale s

JOIN salesperson sp on sp.emp\_no = s.emp\_no

left JOIN dept d on d.dept\_no = sp.dept\_no

manager emp\_no fname lname dept\_no order\_value description

-------------------- ----------- --------------- --------------- ----------- ----------- --------------------------------------------------

Paul Peach 60 Fred Goalie 3 7 Toshiba 6700 Pro

Paul Peach 60 Fred Goalie 3 6 MS Office Professional \* 30

Paul Peach 60 Fred Goalie 3 12 ScanPRO 4800 Scanner

Adam Apricot 10 Alan Brick 1 5 Modems and Cables etc

Paul Peach 60 Fred Goalie 3 2 Laser printer

Paul Peach 50 Ernest Flipper 3 27 Complete Desktop Publishing System

Adam Apricot 10 Alan Brick 1 3 SQL Server 2007 20 user licence

Paul Peach 60 Fred Goalie 3 3 Printer cartridges

--3.2)

SELECT d.manager, sp.emp\_no, sp.fname, sp.lname, sp.dept\_no, s.order\_value, s.description

FROM sale s

JOIN salesperson sp on sp.emp\_no = s.emp\_no

left JOIN dept d on d.dept\_no = sp.dept\_no

where lower(s.description) like '%printer%'

manager emp\_no fname lname dept\_no order\_value description

-------------------- ----------- --------------- --------------- ----------- ----------- --------------------------------------------------

Paul Peach 60 Fred Goalie 3 2 Laser printer

Paul Peach 60 Fred Goalie 3 3 Printer cartridges

--p4) Display each contact's name and their company's name. (10 rows)

-- Make sure you JOIN on the thing they share!!

-- Note how many contacts (1, 2, 3, 3) are in each company as you will 'count' them in code later.

select contact.name, company.name

from contact contact

join company company on contact.company\_no = company.company\_no

Contact\_name Company\_name

-------------------- --------------------

Munching Mike Happy Heaters PLC

Naughty Nick Icicle Igloos Inc

Ollie Octopus Icicle Igloos Inc

Purposeful Peter Judo Jeans PLC

Quentin Quail Judo Jeans PLC

Robber Red Judo Jeans PLC

Ricky Rambo Kipper Kickers Inc

Terrible Tim Kipper Kickers Inc

Uppy Umbrella Kipper Kickers Inc

-- IF YOU HAVE TIME

--p5) Complete this query by filling in the blanks

-- Managers like to telephone contacts after 'big' sales.

-- It's a sort of 'compliance' requirement.

-- A big sale is defined as one where the value of the order is greater than 50% of the

-- salesperson's target, i.e. somebody hits half their yearly target in one deal.

-- The list details the manager, plus the name and telephone number of the contact

-- that the manager must call.

-- You must put in the join predicates and complete the Where clause

.

-- If you are NOT getting a 4 row result set, there is something that MIGHT help you

-- spot the problem.

-- If you are getting 7 rows, try adding the order\_no column to the select list,

-- re-run and look closely.

SELECT manager, C.name, C.tel

FROM dept D INNER JOIN salesperson SP

ON D.dept\_no = sp.dept\_no

INNER JOIN sale S

ON S.emp\_no = SP.emp\_no

INNER JOIN contact C

on C.contact\_code = S.contact\_code

WHERE S.order\_value > .5 \* SP.sales\_target

manager name tel

-------------------- -------------------- -------------------------

Paul Peach Munching Mike (0207)223-9887

Paul Peach Ollie Octopus 0207-341-566670 ext 10

Adam Apricot Munching Mike (0207)223-9887

Paul Peach Purposeful Peter 0131 324545 ext 213

-- Important: When you get this working, can you accurately predict how many rows the result set

-- will contain when you run it WITHOUT the WHERE clause? I.e. natural JOIN of the 4 tables.

-- Did you guess correctly?

-- Answer 8 rows, it is driven by the number of rows in 'sale' (the 'many' table)

-- How many rows would you expect to get if you ran your code after these

-- 3 INSERTs happened?

INSERT INTO dept VALUES (20, 'Dept 20', 'Dept 20 Manager', 20)

INSERT INTO salesperson VALUES (100, 'Pete', 'Pitstop', 2, 5, NULL, NULL, NULL, NULL)

INSERT INTO contact VALUES (4000, 'ZZ', 'Zinedine Zidane', 'Celebrity', NULL, NULL)

-- Answer still 8,

-- adding a dept with no people,

-- a person with no sales,

-- or a contact who has not been sold too

-- will not affect the result of an INNER JOIN with sale

-- Run them (the DELETE statements to remove them are below), and see if you guessed right.

-- Now decide how many rows you would get if you now ran this statement:

INSERT INTO sale

VALUES (900, 10, 1000, 'MM', 3, '05-12-2006', 'Metal 3\*2 Desk')

-- Answer 9 as sale table has 9 rows

-- Guessed correctly? If not ask your instructor.

-- Here are the DELETE statements (run them if you ran any of the 4 INSERTs above).

DELETE FROM salesperson WHERE emp\_no = 100

DELETE FROM dept WHERE dept\_no = 20

DELETE FROM contact WHERE name = 'Zinedine Zidane'

DELETE FROM sale WHERE order\_no = 900

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\* \*\*

\*\* END OF CHAPTER 4. Practical 1 \*\*

\*\* \*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\* \*\*

\*\* CHAPTER 4. Practical 2. \*\*

\*\* Outer JOINs \*\*

\*\* \*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

--p1) This practical is done largely as a 'tutorial' learning exercise, you practise them later.

-- Run the following 3 precoded queries.

-- NOTE - there are 5 depts with 5 managers.

-- NOTE - the salespeople are in 3 different depts.

-- NOTE - the 3rd query fails to list 'Diver Dan' or 'Xavier Xylophone' because they each manage a 'dept' with no people.

SELECT dept\_no, manager

FROM dept

SELECT DISTINCT dept\_no AS 'Distinct list of depts that people are in'

FROM salesperson

SELECT D.dept\_no, manager, lname

FROM salesperson SP

INNER JOIN dept D ON SP.dept\_no = D.dept\_no

dept\_no manager

----------- --------------------

1 Adam Apricot

2 Barbara Banana

3 Paul Peach

4 Diver Dan

5 Xavier Xylophone

Distinct list of depts that people are in

-----------------------------------------

1

2

3

dept\_no manager lname

----------- -------------------- ---------------

1 Adam Apricot Brick

2 Barbara Banana Custard

2 Barbara Banana Digger

3 Paul Peach Ernst

3 Paul Peach Flipper

3 Paul Peach Goalie

--p2) Copy in the 3rd query from p1) and change the word INNER to the word RIGHT

-- This is called an outer JOIN, inserting the word RIGHT means literally

-- "include every row from the table on the RIGHT of the word JOIN",

-- (even if there is no matching row in the table on the LEFT).

SELECT D.dept\_no, manager, lname

FROM salesperson SP

RIGHT JOIN dept D ON SP.dept\_no = D.dept\_no

dept\_no manager lname

----------- -------------------- ---------------

1 Adam Apricot Brick

2 Barbara Banana Custard

2 Barbara Banana Digger

3 Paul Peach Ernst

3 Paul Peach Flipper

3 Paul Peach Goalie

4 Diver Dan NULL

5 Xavier Xylophone NULL

-- Now 'Coalesce' the lname column to display 'Nobody in this dept' where appropriate.

SELECT D.dept\_no, manager, COALESCE(SP.lname, 'Nobody in this department') as Surname

FROM salesperson SP RIGHT JOIN dept D

ON SP.dept\_no = D.dept\_no

dept\_no manager Surname

----------- -------------------- -------------------

1 Adam Apricot Brick

2 Barbara Banana Custard

2 Barbara Banana Digger

3 Paul Peach Ernst

3 Paul Peach Flipper

3 Paul Peach Goalie

4 Diver Dan Nobody in this dept

5 Xavier Xylophone Nobody in this dept

-- Recognize that

FROM dept D LEFT JOIN salesperson SP

-- would give the same result as

FROM salesperson SP RIGHT JOIN dept D

-- p3) Ask yourself whether this FROM clause makes any sense?

FROM dept D RIGHT JOIN salesperson SP

-- Does this translate into

-- "Show me all the people even one's in a non existent dept"?

-- Hopefully the referential integrity (covered fully later) between the tables

-- will ensure that if you have depts 1-5 only, that there is no one in dept 6 or 7 or 87.

-- But can a salesperson be in no (NULL) dept at all?

-- Well, it depends whether 'dept\_no' of 'salesperson' is a NULLable (optional) column or not.

-- In your schema/table it IS an optional column so the following INSERT (try it) will run ok.

INSERT INTO salesperson(emp\_no, fname, lname, dept\_no)

VALUES (70, 'Monica', 'Ell', NULL)

-- The following INNER JOIN query will not discover her

SELECT manager, COALESCE(lname, 'Nobody in this dept') AS Surname

FROM salesperson SP INNER JOIN dept D

ON SP.dept\_no = D.dept\_no

manager Surname

-------------------- -------------------

Adam Apricot Brick

Barbara Banana Custard

Barbara Banana Digger

Paul Peach Ernst

Paul Peach Flipper

Paul Peach Goalie

-- Nor will this Outer JOIN

SELECT manager, COALESCE(lname, 'Nobody in this dept') AS Surname

FROM salesperson SP RIGHT JOIN dept D

ON SP.dept\_no = D.dept\_no

manager Surname

-------------------- -------------------

Adam Apricot Brick

Barbara Banana Custard

Barbara Banana Digger

Paul Peach Ernst

Paul Peach Flipper

Paul Peach Goalie

Diver Dan Nobody in this dept

Xavier Xylophone Nobody in this dept

-- But this one will, note dept D RIGHT JOIN salesperson SP

SELECT COALESCE(manager,'Has no manager') AS Manager, lname

FROM dept D RIGHT JOIN salesperson SP

ON SP.dept\_no = D.dept\_no

Manager lname

-------------------- ---------------

Adam Apricot Brick

Barbara Banana Custard

Barbara Banana Digger

Paul Peach Ernst

Paul Peach Flipper

Paul Peach Goalie

Has no manager Ell

-- p4) ‘Can I do a LEFT & a RIGHT JOIN at the same time?’ is a question often asked.

-- Yes, it’s called a FULL JOIN.

SELECT COALESCE(manager, '\*\* Has no manager \*\*') AS Manager,

COALESCE(lname, '\*\* Nobody in this dept \*\*') AS Surname

FROM dept D FULL JOIN salesperson SP

ON SP.dept\_no = D.dept\_no

Manager Surname

-------------------- -------------------------

Adam Apricot Brick

Barbara Banana Custard

Barbara Banana Digger

Paul Peach Ernst

Paul Peach Flipper

Paul Peach Goalie

Diver Dan \*\* Nobody in this dept \*\*

Xavier Xylophone \*\* Nobody in this dept \*\*

\*\* Has no manager \*\* Ell

-- But if you were now to include 'dept\_no' in this query, should you choose

-- SP.dept\_no or D.dept\_no, because one of the rows is surely going to have a NULL dept\_no?

-- Well you could include either and COALESCE it to remove the NULL, but there is one problem.

-- Namely, the 'COALESCE' function requires that both parameters are of the same data type,

-- so

COALESCE(D.dept\_no, 0 ) -- would be valid syntax but produce a misleading '0' in results.

COALESCE(D.dept\_no, 'n/a') -- would fail as 1st arg is numeric, but 2nd arg is not.

-- So the solution would be to convert the 'dept\_no' into a character string.

-- In SQL Server it would be

COALESCE(STR(D.dept\_no, 2), 'n/a') -- meaning convert D.dept\_no into a 2 character string.

-- In Oracle it would be

NVL(TO\_CHAR(D.dept\_no, '99'), 'n/a')

-- or

COALESCE(TO\_CHAR(D.dept\_no, '99'), 'n/a')

SELECT COALESCE(STR(D.dept\_no,2),'N/A') AS Dept\_no,

COALESCE(manager,'\*\* Has no manager \*\*') AS Manager,

COALESCE(lname,'\*\* Nobody in this dept \*\*') AS Surname

FROM dept D FULL JOIN salesperson SP

ON SP.dept\_no = D.dept\_no

Dept\_no Manager Surname

------- -------------------- -------------------------

1 Adam Apricot Brick

2 Barbara Banana Custard

2 Barbara Banana Digger

3 Paul Peach Ernst

3 Paul Peach Flipper

3 Paul Peach Goalie

4 Diver Dan \*\* Nobody in this dept \*\*

5 Xavier Xylophone \*\* Nobody in this dept \*\*

N/A \*\* Has no manager \*\* Ell

-- p5) The odd thing is, what happens when you add the following 'WHERE' clause to

-- this 'Left' JOIN.

SELECT D.dept\_no,

COALESCE(manager,'\*\* Has no manager \*\*') AS Manager,

COALESCE(lname,'\*\* Nobody in this dept \*\*') AS Surname

FROM dept D LEFT JOIN salesperson SP

ON SP.dept\_no = D.dept\_no

WHERE SP.dept\_no IS NULL

dept\_no Manager Surname

----------- -------------------- -------------------------

4 Diver Dan \*\* Nobody in this dept \*\*

5 Xavier Xylophone \*\* Nobody in this dept \*\*

-- You have just seen a technique (there are others, as we will see) of how to find the

-- depts that have no salespeople by JOINing Dept to Salesperson,

-- including all the 'spares', and then just retaining the 'spares'.

-- Later in the course you will see 2 other ways of finding 'Depts with no People'.

-- The mistake you must not make is to write this 'WHERE' clause:

WHERE SP.post\_code IS NULL -- Why?

-- Because then you will be listing managers of depts (and their employees) that EITHER

-- a) have no people in them, or

-- b) have a person, but a person who has no post\_code.

SELECT D.dept\_no,

COALESCE(manager,'\*\* Has no manager \*\*') AS Manager,

COALESCE(lname,'\*\* Nobody in this dept \*\*') AS Surname

FROM dept D LEFT JOIN salesperson SP

ON SP.dept\_no = D.dept\_no

WHERE SP.post\_code IS NULL -- no good, must be a primary key(not NULL) column for safety

dept\_no Manager Surname

----------- -------------------- -------------------------

3 Paul Peach Ernst

3 Paul Peach Goalie

4 Diver Dan \*\* Nobody in this dept \*\*

5 Xavier Xylophone \*\* Nobody in this dept \*\*

-- p6) Clean up, DELETE that 7th employee you added

DELETE FROM salesperson WHERE emp\_no = 70

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\* \*\*

\*\* END OF CHAPTER 4. Practical 2. \*\*

\*\* \*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/