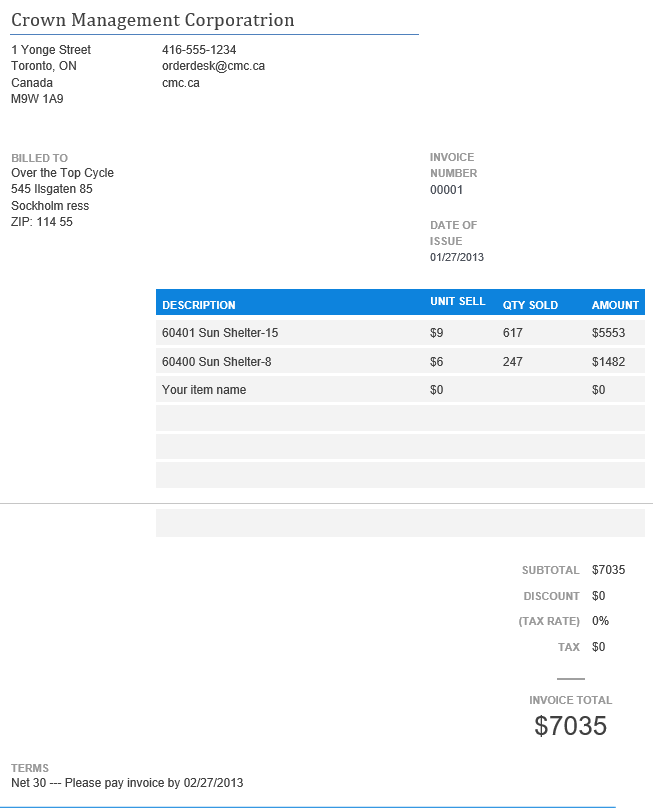
Shortened Notes on Joins

1) Here is a sample invoice for customer 1085, order number 1 with 2 products ordered.

This is the selling company data



This due date is calcuated by order\_dt + 1 month

The AMOUNT column and totals are calculated fields and are generally not stored. The total value goes to another table under the Accounts Receivable department showing what the customer owes.

Each of these 2 lines is one row in the ORDERLINES table.

The product name and sell price are from the PRODUCTS table

Order\_no and order\_dt from ORDERS

Customer data from CUSTOMERS table

This demonstrates that in order to do an order/invoice many tables need to be joined.

The same for a lot of processes in the company.

Simplifying joins

EQUI-JOINS (also called INNER JOINS)

To get customer id, customer name, order number and order\_date. You need to look at what tables hold this data. … CUSTOMERS hold the name and number, ORDERS hold the order\_no, date and customer number as an FK to customer. Since I don't want to see all the orders from the customer, I am limiting it to order\_no = 1

**SELECT c.cust\_no, cname, order\_no, order\_dt**

**FROM customers c ,orders o**

**WHERE c.cust\_no = o.cust\_no**

**AND o.order\_no = 1;**

CUST\_NO CNAME ORDER\_NO ORDER\_DT

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

1085 Over the Top Cycles 3 AB 1 27-Jan-2013

The improved way using JOIN ON

**SELECT c.cust\_no, cname, order\_no, order\_dt**

**FROM customers c JOIN orders o**

Here are the join conditions

**ON (c.cust\_no = o.cust\_no)**

**WHERE o.order\_no = 1;**

And this is the WHERE condition that limits the rows to look at.

TO DO THE INVOICE will require joining many tables

THE ABOVE IS THE MOST COMMON JOIN TYPE

OUTER JOINS

You are running a little kids baseball club that every spring needs to register children into teams. To keep it very simple you have 2 tables called TEAMS and KIDS

TEAMS [teamid, temaname]

KIDS [kidid, kidname, teamid] The teamid starts as null until the child is assigned to a team.

PROBLEM to solve:

After 5 days the CEO of the baseball league asks for

Give me the team name and kid name in that team.

ANSWER: Using Equi-joins

SELECT teamname, kidname

FROM kids k, teams t

WHERE k.teamid = t.teamid;

Better method:

SELECT teamname, kidname

FROM kids k, JOIN teams t

ON (k.teamid = t.teamid);

The result is a listing of teams that have kids assigned to them already. However, if there was not a match based on the WHERE then some teams were not mentioned and some kids who registered but were not assigned a team, will not appear as output.

Show me ALL teams and any kid assigned to them

OUTER JOINS

SELECT teamname, kidname

FROM kids k, teams t

WHERE k.teamid = t.**teamid (+); 🡸 very old method**

BETTER METHOD ….

SELECT teamname, kidname

FROM kids k RIGHT JOIN teams t 🡸 RIGHT because TEAM is on the right

ON k.teamid = t.teamid

Show me teams with kids and any kid not assigned to a team

SELECT teamname, kidname

FROM kids k LEFT JOIN teams t 🡸 LEFT because KIDS is on the left

ON k.teamid = t.teamid

SELECT teamname, kidname

FROM teams t RIGHT JOIN kids k 🡸 it is RIGHT, want all kids. We just changed table order

ON k.teamid = t.teamid

Show me all teams and all kids even if the kids are not yet matched up to a team. You want left and right side.

SELECT teamname, kidname

FROM kids k FULL JOIN teams t

ON k.teamid = t.teamid

SELF JOIN

Sometimes (usually it occurs with employee tables) you need to join a copy of the table to itself.

I want a list of manager names and the employees that work for them. Because managers and non-managers are all employees there is no need for a separate manager table.

Notice that all employees have the data about name, address etc and also an ID pointing to who their manager is

30 Lee Chan other info … 149 (manager id) 80(dept id)

To know Chan's manager, we look for manger id 149 and match it to employee 149 to get the manager

SELECT m.last\_name, e.last\_name

FROM employees e JOIN employees m

ON e.manager\_id = m.employee\_id;

Partial result shown out of 53 rows

LAST\_NAME LAST\_NAME

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

De Haan Hunold

Hartstein Fay

Higgins Gietz

Hunold Lorentz

Hunold Ernst

King Zlotkey

King Mourgos

Kochhar Whalen

Kochhar Higgins

Mourgos Vargas

Mourgos Rajs

Mourgos Matos

Mourgos Davies

Zlotkey Chancevente

**Zlotkey Chan**

Zlotkey Brigade

53 rows selected

Better to label the columns as to who is a manager and who is not

NOW READ NOTES ON NON-EQUI JOINS