**Joins Procedures and Triggers**

--1

SELECT p.pub\_name AS 'Publisher Name',

t.title AS 'Book Name',

s.qty AS 'Quantity',

s.qty \* t.price AS 'Sale Amount',

a.au\_fname + ' ' + a.au\_lname AS 'Author Name'

FROM publishers p

JOIN titles t ON p.pub\_id = t.pub\_id

JOIN sales s ON t.title\_id = s.title\_id

JOIN titleauthor ta ON t.title\_id = ta.title\_id

JOIN authors a ON ta.au\_id = a.au\_id;

\*To print books that haven't been sold, you can use a query like this:

SELECT t.title AS 'Book Name'

FROM titles t

LEFT JOIN sales s ON t.title\_id = s.title\_id

WHERE s.title\_id IS NULL;

\*To print authors who haven't written any books, you can use a query like this:

SELECT a.au\_fname + ' ' + a.au\_lname AS 'Author Name'

FROM authors a

LEFT JOIN titleauthor ta ON a.au\_id = ta.au\_id

WHERE ta.title\_id IS NULL;

--2

CREATE PROCEDURE GetTotalSalesForAuthor(@authorName VARCHAR(100))

AS

BEGIN

DECLARE @totalSalesAmount DECIMAL(10, 2)

-- Calculate the total sales amount for books authored by the given author

SELECT @totalSalesAmount = SUM(s.qty \* t.price)

FROM titleauthor ta

JOIN authors a ON ta.au\_id = a.au\_id

JOIN titles t ON ta.title\_id = t.title\_id

LEFT JOIN sales s ON t.title\_id = s.title\_id

WHERE a.au\_fname + ' ' + a.au\_lname = @authorName

-- Check if there are no sales

IF @totalSalesAmount IS NULL

BEGIN

PRINT 'Sale yet to gear up'

END

ELSE

BEGIN

PRINT 'Total Sales Amount for ' + @authorName + ': $' + CAST(@totalSalesAmount AS VARCHAR(20))

END

END

--3

SELECT s.\*

FROM sales s

WHERE s.qty > ALL (

SELECT MAX(same\_title\_sale.qty)

FROM sales same\_title\_sale

WHERE same\_title\_sale.title\_id = s.title\_id

AND same\_title\_sale.stor\_id = s.stor\_id

);

--4

SELECT CONCAT(a.au\_fname, ' ', a.au\_lname) AS 'Author Name',

AVG(t.price) AS 'Average Price'

FROM authors a

JOIN titleauthor ta ON a.au\_id = ta.au\_id

JOIN titles t ON ta.title\_id = t.title\_id

GROUP BY a.au\_fname, a.au\_lname;

--5

sp\_columns 'titles';

EXEC sp\_columns 'titles';

SELECT constraint\_name, constraint\_type

FROM information\_schema.table\_constraints

WHERE table\_name = 'titles';

--6

CREATE PROCEDURE GetBookCountPricedLessThan(@inputPrice DECIMAL(10, 2))

AS

BEGIN

DECLARE @bookCount INT

-- Calculate the count of books priced less than the input price

SELECT @bookCount = COUNT(\*)

FROM titles

WHERE price < @inputPrice

PRINT 'Number of books priced less than $' + CAST(@inputPrice AS VARCHAR(10)) + ': ' + CAST(@bookCount AS VARCHAR(10))

END

EXEC GetBookCountPricedLessThan 20.00;

--7

ALTER TABLE titles

ADD CONSTRAINT CHK\_PriceAboveSeven CHECK (price >= 7);

--8

SELECT title

FROM titles

WHERE title LIKE '%e%' AND title LIKE '%a%';