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

\*\* \*\*

\*\* CHAPTER 6. DDL Demo. \*\*

\*\* CREATE/ALTER Table Statement \*\*

\*\* Defining Primary Keys and Foreign Keys using DRI, \*\*

\*\* and observing INSERT / UPDATE / DELETE behaviour \*\*

\*\* \*\*

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

--p1) Run the following syntax to CREATE a 'review' table:

Use database0

CREATE TABLE review

(

emp\_no INT NOT NULL,

review\_date DATETIME NOT NULL,

grade CHAR(1) NULL

)

-- Try running it again.

--p2) Add a 4th column called 'notes'; it is to be a character column of varying length, not

-- to exceed 40 characters in length and is to allow NULLs.

ALTER TABLE review

ADD notes VARCHAR(40) NULL

-- Try running it again.

--p3) Inspect the table structure within the system tables. (SQL SERVER specific code supplied)

sp\_help review

--p4) INSERT two rows into the table. The values are duplicated, but it will

-- run quite succesfully because the 'review' table has no primary key defined and

-- therefore no uniqueness is being enforced.

INSERT INTO review

VALUES (23, '05-12-2006', 'A','He is happy')

INSERT INTO review

VALUES (23, '09-30-2006', 'A','He is still happy')

--p5) Display the data in the table

SELECT \*

FROM review

emp\_no review\_date grade notes

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

23 2006-05-12 00:00:00.000 A He is happy

23 2006-09-30 00:00:00.000 A He is still happy

-- So 2 reviews for employee 23 have been added, but employee 23 does not exist in

-- salesperson. Your database is lacking in 'referential integrity', because we never told

-- the DBMS about any relationship between emp\_no of 'review' and emp\_no of 'salesperson'.

-- Let's do something about that!

--p6) Drop the 'review' table

DROP TABLE review

--p7) Now we will reCREATE 'review' table this time with some additional constraints.

-- There is a composite Primary Key constraint defined on the 'emp\_no' and 'review\_date'

-- columns. The constraint is a separate object in the database which can be 'dropped'

-- independently of the table itself, so it is a good idea to give it a meaningful name

-- like 'pk\_review'.

CONSTRAINT pk\_review PRIMARY KEY(emp\_no, review\_date)

-- There is a Foreign Key constraint defined on 'emp\_no' which basically says

-- "values in the 'emp\_no' column of 'review' should exist in the primary key column

-- of 'salesperson' which just happens to also be called 'emp\_no' but doesn’t have to be".

CONSTRAINT fk\_salesperson\_link FOREIGN KEY(emp\_no) REFERENCES salesperson

-- Now run this CREATE table syntax and then observe the changed behaviour you get when you

-- either INSERT/UPDATE 'reviews' or DELETE/UPDATE 'salespersons'.

-- Run this code:

CREATE TABLE review

(

emp\_no INT NOT NULL,

review\_date DATETIME NOT NULL,

grade CHAR(1) NULL,

CONSTRAINT pk\_review PRIMARY KEY(emp\_no, review\_date),

CONSTRAINT fk\_salesperson\_link FOREIGN KEY(emp\_no) REFERENCES salesperson

)

--p8) Run all the following code samples one by one observing the output and explaining

-- briefly, ‘why?’.

-- First, try to add a row to the review table, for employee 40. (code supplied)

INSERT INTO review

VALUES (40, '05-12-2006', 'A')

-- What happened and why?

-- It worked but only because 'emp\_no=40' does exist in 'salesperson'.

--p9) Try to add a row for employee 90, who does not exist as a salesperson. (code supplied)

INSERT INTO review

VALUES (90, '05-12-2006', 'A')

-- What happened and why?

INSERT statement conflicted with COLUMN FOREIGN KEY constraint 'fk\_salesperson\_link'.

The conflict occurred in database 'xxxxxxxxxx', table 'salesperson', column 'emp\_no'.

The statement has been terminated.

-- INSERT failed

--p10) The following statements all violate the referential integrity between the 2 tables.

UPDATE review

SET emp\_no = 41

WHERE emp\_no = 40

DELETE FROM salesperson

WHERE emp\_no = 40

UPDATE salesperson

SET emp\_no = 41

WHERE emp\_no = 40

-- all fail for same reason.

--p11) Decide what should happen from 2 suggested answers if you were to run the next statement.

DELETE FROM dept -- NOTE no WHERE clause!!

-- To remind you: Dept nos are currently 1, 2, 3 & 4 but there are no people in dept 4.

-- So should it therefore ?

-- a) DELETE just dept number 4 and leave the other 3 depts intact so as not to orphan

-- the employees in them, or

-- b) do nothing, as it cannot achieve everything that you intended to do in this

-- statement.

a

-- Try it

-- But,

DELETE FROM sale

-- would run just fine as nothing depends on a sale.

-- The following statements will reinstate your sale table if you run the 'DELETE FROM sale'.

INSERT INTO sale VALUES(100, 60, 1000, 'MM', 7.00, '06-24-2006', 'Toshiba 6700 Pro' )

INSERT INTO sale VALUES(200, 60, 3000, 'QQ', 6.00, '05-01-2006', 'MS Office Professional \* 30' )

INSERT INTO sale VALUES(300, 60, 2000, 'OO', 12.00, '07-14-2006', 'ScanPRO 4800 Scanner' )

INSERT INTO sale VALUES(400, 10, 1000, 'MM', 5.00, '08-09-2006', 'Modems and Cables etc' )

INSERT INTO sale VALUES(500, 60, 4000, 'TT', 2.00, '07-23-2006', 'Laser printer' )

INSERT INTO sale VALUES(600, 50, 3000, 'PP', 27.00, '05-23-2006', 'Complete Desktop Publishing System' )

INSERT INTO sale VALUES(700, 10, 2000, 'OO', 3.00, '01-23-2006', 'SQL Server 2007 20 user licence' )

INSERT INTO sale VALUES(800, 60, 3000, 'RR', 3.00, '11-15-2006', 'Printer cartridges' )

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

\*\* \*\*

\*\* END OF CHAPTER 6. Demo. \*\*

\*\* \*\*

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