-- A report query that uses a JOIN (any type) to report on some aggregate value based on a group by clause.

SELECT crew\_assignment.EMP\_ID,

SUM(TRUNCATE(TIMESTAMPDIFF(MINUTE,flight.fly\_departure\_dt,flight.fly\_arrival\_dt)/60,2)) AS 'HOURS FLOWN'

FROM crew\_assignment

JOIN flight

WHERE crew\_assignment.FLY\_ID = flight.FLY\_ID

AND TIMESTAMPDIFF(MINUTE,flight.fly\_departure\_dt,flight.fly\_arrival\_dt) > 0

GROUP BY crew\_assignment.EMP\_ID

ORDER BY crew\_assignment.EMP\_ID;

-- A query that includes a subquery.

SELECT flight.FLY\_ID, (AMOD\_FIRST + AMOD\_ECONPLUS + AMOD\_BUSINESS + AMOD\_ECON) AS 'TOTAL\_SEATS',

(SELECT COUNT(BOOK\_SEATNUM) FROM booking WHERE BOOK\_FLY\_ID = flight.FLY\_ID) AS 'BOOKED\_SEATS'

FROM (flight

INNER JOIN aircraft ON flight.AIRCRAFT\_AIRCRAFT\_ID = aircraft.AIRCRAFT\_ID

INNER JOIN aircraft\_model ON aircraft.AMOD\_ID = aircraft\_model.AMOD\_ID)

ORDER BY flight.FLY\_ID;

-- A query that creates a view.

CREATE VIEW current\_employees AS

SELECT employee.EMP\_LNAME, employee.EMP\_FNAME, job.JOB\_NAME, job.JOB\_DEPT

FROM employee

JOIN job ON employee.JOB\_ID = job.JOB\_ID

WHERE EMP\_VALID\_EMP = 1

ORDER BY JOB\_DEPT ASC, JOB\_NAME ASC, EMP\_LNAME ASC;

SELECT \* FROM current\_employees;

-- A query that demonstrates the view's use.

SELECT \*

FROM current\_employees

WHERE JOB\_DEPT = 'Airport Operations';

-- A stored procedure that can be called by a query to perform some math operation on the data and returns a value(s).

delimiter //

CREATE PROCEDURE sp\_flightSeats()

BEGIN

SELECT flight.FLY\_ID, (AMOD\_FIRST + AMOD\_ECONPLUS + AMOD\_BUSINESS + AMOD\_ECON) AS 'TOTAL\_SEATS',

(SELECT COUNT(BOOK\_SEATNUM) FROM booking WHERE BOOK\_FLY\_ID = flight.FLY\_ID) AS 'BOOKED\_SEATS',

(AMOD\_FIRST + AMOD\_ECONPLUS + AMOD\_BUSINESS + AMOD\_ECON) -

(SELECT COUNT(BOOK\_SEATNUM) FROM booking WHERE BOOK\_FLY\_ID = flight.FLY\_ID) AS 'EMPTY\_SEATS'

FROM (flight

INNER JOIN aircraft ON flight.AIRCRAFT\_AIRCRAFT\_ID = aircraft.AIRCRAFT\_ID

INNER JOIN aircraft\_model ON aircraft.AMOD\_ID = aircraft\_model.AMOD\_ID)

ORDER BY flight.FLY\_ID;

END // delimiter ;

CALL sp\_flightSeats();

-- A stored procedure that uses a cursor to access and manipuate (update/change) data.

delimiter //

CREATE PROCEDURE sp\_updateContact(employeeID int UNSIGNED, lastName varchar(45), firstName varchar(45), initial char(1), phoneNumber varchar(15))

BEGIN

DECLARE completed int default 0;

DECLARE ContactEmpID int UNSIGNED default 0;

DECLARE myCursor CURSOR FOR SELECT EMCT\_EMP\_ID FROM emergency\_contact LIMIT 1;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET completed = 1;

OPEN myCursor;

get\_contact: LOOP

FETCH myCursor INTO ContactEmpID;

IF completed = 1 THEN

LEAVE get\_contact;

ELSEIF ContactEmpID = employeeID THEN

UPDATE g3ds.emergency\_contact

SET EMCT\_LNAME = lastName,

EMCT\_FNAME = firstName,

EMCT\_MID\_INIT = initial,

EMCT\_PHONE\_NUM = phoneNumber

WHERE EMCT\_EMP\_ID = employeeID;

END IF;

END LOOP get\_contact;

CLOSE myCursor;

END // delimiter ;

CALL sp\_updateContact(100000, 'Ono', 'Yoko', null, '914-237-0134');

-- A trigger that updates/inserts data based on an insert.

delimiter //

CREATE PROCEDURE sp\_addContact(employeeID INT UNSIGNED)

BEGIN

INSERT INTO emergency\_contact (EMCT\_EMP\_ID, EMCT\_LNAME, EMCT\_FNAME, EMCT\_MID\_INIT, EMCT\_PHONE\_NUM)

VALUES (employeeID, '', '', null, '');

END

// delimiter ;

delimiter //

CREATE TRIGGER add\_contact AFTER INSERT ON employee

FOR EACH ROW

CALL sp\_addContact(NEW.EMP\_ID);

// delimiter ;

-- A query that demos the trigger.

INSERT INTO employee

VALUES(100000, 'Lennon', 'John', 'W', CAST('1940-10-09' AS DATE), 'W 72nd St', 'NY', '10023', '914-237-0133', 1, 'New York', 21);

SELECT \*

FROM emergency\_contact

WHERE EMCT\_EMP\_ID = 100000;