1. Workers and production managers, and producers are our users

D2 - Hiring a new worker/Insert new worker into workers

INSERT INTO Worker (DepartmentID, Name, SIN, Occupation, HourlyRate, UnionID

VALUES (2, “Daniel Brodsky”, “dog jerker”, 13253121, 10, 3)

D3 - Firing or laying off a worker/Delete worker from workers

DELETE FROM Worker

WHERE employee\_id = 200

D4 - Employee gets a raise/Worker hourly rate gets updated

UPDATE Worker

SET HourlyRate = 20

WHERE employee\_id = 200

D5 - Find all people names of one occupation working on a given movie/join Movie to Works on to Worker

SELECT W.Name

FROM Worker W, WorkerOn, O, Movie M

WHERE W.WorkerID = O.WorkerID AND O.ProductionID = M.ProductionID AND Movie =

“MOVIE TITLE” AND W.Occupation = “OCCUPATION”

**\*We have selected all timetables signed by a certain manager instead**

SELECT w.\* FROM employee w, timetable c, m\_signs\_t m WHERE m.manager\_id = 500 AND m.timetable\_id = c.timetable\_id AND c.employee\_id = w.employee\_id

D6 - Find timetables submitted by one crew member/join worker to Timetable

SELECT W.\*

FROM Worker W, TimeTable T

WHERE W.WorkerID = TimeTable.WorkerID

D7 - Find all payments created by a manager/join manager to payment

SELECT P.\*

FROM Payment P, Manager M

WHERE M.ManagerID = P.PaymentID

**\*We replaced that with find all payments received by one employee in their employee page**

SELECT ee.name as mname, p.\* FROM payment p, employee e, manager m INNER JOIN employee ee on' **+**

' m.employee\_id = ee.employee\_id WHERE e.employee\_id = ' **+** employee\_id **+**

' and p.employee\_id = ' **+** employee\_id **+** ';'

D8 - Separate between workers by their department/group workers by department name

(Count the amount of workers in each department)

SELECT count(w.DepartmentID) AS PeopleInDept, d.DepartmentName

FROM Worker w, Department d

ORDER BY w.DepartmentID

D9 - select attributes from worker

SELECT \*

FROM Worker w

WHERE w.WorkerID = “SOME ID”

**\*we have select all from manager instead**

SELECT \* FROM manager;

D10 - select all workers in a given movie

SELECT w.\*

FROM Worker w, Movie m, WorkersOn o,

WHERE w.WorkerID = o.WorkerID AND o.ProductionID = m.ProductionID AND m.ProductionTitle = “SOME TITLE”

**\*We have selected all calls originating from a certain manager instead**

SELECT ee.name as mname, c.\*, e.\* FROM call\_sheet c, employee e,' **+**

' manager m INNER JOIN employee ee on m.employee\_id = ee.employee\_id WHERE e.employee\_id = ' **+**

employee\_id **+** ' and c.employee\_id = ' **+** employee\_id **+** ' and c.manager\_id = m.manager\_id;

D11 - select all payments within a given time

SELECT \*

FROM Payment

WHERE P.ChequeDate > “2013-01-01” AND P.ChequeDate < “2014-01-01”

**\*We replaced that with select all timetables involving a certain employee**

'SELECT t.\* FROM timetable t, employee e WHERE e.employee\_id = ' **+** employee\_id **+**' and t.employee\_id = ' **+** employee\_id **+** ';'

D12 -

Crew member sees Production title from Movie, Department name from Department, all columns from worker table, sees all columns in callsheet, payment and timetable of that crew member

Production manager sees all columns for the workers who are part of the same production and department

Producer sees all columns for the workers who are part of the same production

**CREATE** **VIEW** Current\_payments **as** (

**SELECT** **\***

**FROM** Payment

**WHERE** date\_part('year', Payment\_date) **=** date\_part('year', **CURRENT\_DATE**)

);

**CREATE** **VIEW** Current\_call\_sheets **as** (

**SELECT** **\***

**FROM** Call\_sheet

**WHERE** date\_part('year', **Time**) **=** date\_part('year', **CURRENT\_DATE**)

);

D13 –

Create tables and db objects in SQL – Daniel

Create data for tables – James

Code the queries and test – Osman, James

Graphical user interface – Minxing, Daniel

Testing and error handling – Osman

Documentation – equal split