### EMP:

|  |  |  |
| --- | --- | --- |
| ENO | ENAME | TITLE |
| E1 | J. Doe | Elect. Eng. |
| E2 | M. Smith | Sys. Anal. |
| E3 | A. Lee | Mech. Eng. |
| E4 | J. Miller | Programmer |
| E5 | B. Casey | Sys. Anal. |
| E6 | L. Chu | Elect. Eng. |
| E7 | R. Davis | Mech. Eng. |
| E8 | J. Jones | Sys. Anal. |

### ASG:

|  |  |  |  |
| --- | --- | --- | --- |
| ENO | PNO | RESP | DUR |
| E1 | P1 | Manager | 12 |
| E2 | P1 | Analysist | 24 |
| E2 | P2 | Analysist | 6 |
| E3 | P3 | Consultant | 10 |
| E3 | P4 | Engineer | 48 |
| E4 | P2 | Programmer | 18 |
| E5 | P2 | Manager | 24 |
| E6 | P4 | Manager | 48 |
| E7 | P3 | Engineer | 36 |
| E8 | P3 | Manager | 40 |

### PROJ:

|  |  |  |  |
| --- | --- | --- | --- |
| PNO | PNAME | BUDGET | LOC |
| P1 | Instrumentation | 150,000 | Montreal |
| P2 | Database Develop. | 135,000 | New York |
| P3 | CAD/CAM | 250,000 | New York |
| P4 | Maintenance | 310,000 | Paris |

### PAY:

|  |  |
| --- | --- |
| TITLE | SAL |
| Elect. Eng. | 40,000 |
| Sys. Anal. | 34,000 |
| Mech. Eng. | 27,000 |
| Programmer | 24,000 |

# Problem 5.2

**Application 1:**

Del data opp mellom forskjellige RESP verdier:

* RESP = 'Manager'
* RESP = 'Analasist'
* RESP = 'Consultant'
* RESP = ' Engineer '
* RESP = ' Programmer '

**Application 2:**

Del opp mellom <20 og >=20 DUR verdier:

* DUR < 20
* DUR >= 20

**Vi kan kombinere disse slik at vi får DHFen:**

F1: (Manager, DUR < 20)

F2: (Manager, DUR >= 20)

F3: (Analasist, DUR < 20)

F4: (Analasist, DUR >= 20)

F5: (Consultant, DUR < 20)

F6: (Consultant, DUR >= 20)

F7: (Engineer, DUR < 20)

F8: (Engineer, DUR >= 20)

F9: (Programmer, DUR < 20)

F10: (Programmer, DUR >= 20)

# Problem 5.8 (VER 1)

**Application 1:**

*CREATE VIEW EMPVIEW (ENO, ENAME, TITLE, PNO, RESP) =*

*AS SELECT EMP.ENO, EMP. ENAME, EMP. TITLE, ASG.PNO, ASG.RESP*

*FROM EMP, ASGWHERE EMP.ENO=ASG.ENO AND ASG.DUR=24*

*AND EMP.TITLE=”Programmer”*

**Application 2:**

*SELECT ENO, DUR*

*FROM ASG*

**Fra dette har vi variablene:**

**Vi lager en use(qi, Aj) matrise**

**Vi har også matrisen use(qi,Si)**

**Vi kan nå lage en affinity matrix ved å kalkulere:**

For alle verdier n og i=[2, 3, 5, 6]:

For n=4 og i=[2, 3, 5, 6]:

**Fra dette lager vi en affinity matrix**

**We use heuristics**

We see here that are all tightly connected. Furthermore, we see that is quite isolated. Therefore it would be smart to separate this into it’s own database. I would therefore propose the following setup:

# Problem 5.8 (VER 2)

**Application 1:**

*CREATE VIEW EMPVIEW (ENO, ENAME, TITLE, PNO, RESP) =*

*AS SELECT EMP.ENO, EMP. ENAME, EMP. TITLE, ASG.PNO, ASG.RESP*

*FROM EMP, ASG WHERE EMP.ENO=ASG.ENO AND ASG.DUR=24*

*AND EMP.TITLE=”Programmer”*

**Application 2:**

*SELECT ENO, DUR*

*FROM ASG*

**Fra dette har vi variablene:**

**Vi lager en use(qi, Aj) matrise**

**Vi har også matrisen use(qi,Si)**

**Vi kan nå lage en affinity matrix ved å kalkulere:**

For alle verdier n=i=[1,2,3,5,6]

For 4 og 7:

**Fra dette lager vi en affinity matrix**

**Ved å flytte på dette får vi:**

**We use heuristics**

We see here that are tightly connected. Therefore it would be smart to perform vertical separation with these in mind. I would therefore propose the following vertical fragmentation: