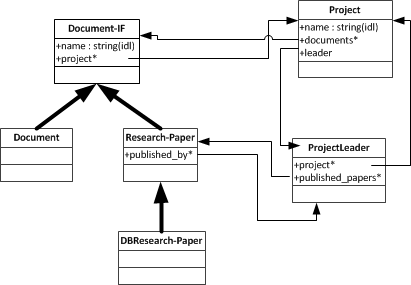
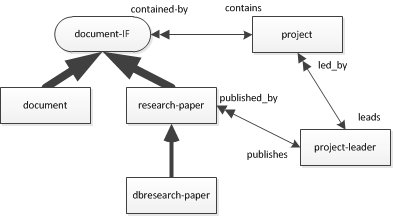
1.

a)



b)



2.

DocumentIF subclass 'ResearchPaper'

instVarNames: #( 'name', 'published\_by' )

inDictionary: UserGlobals

constraints: #[

#[#name, String],

#[published\_by, ProjectLeader]

]

classmethod:ResearchPaper newName:aName, newPublisher:publisher

|tempPaper|

tempPaper:= self new

tempPaper name:aName published\_by:publisher

^tempPaper

.

ResearchPaper subclass 'DBResearchPaper'

instVarNames: #( 'db\_type' )

inDictionary: UserGlobals

constraints #[

#[#db\_type, String]

]

classmethod:DBResearchPaper newName:aName, newPublisher:publisher, newType:type

|tempPaper|

tempPaper:= super newName:aName, newPublisher:publisher

tempPaper db\_type:type

^tempPaper

.

3.

a) SELECT ProjectName: P.name

FROM Projects P

WHERE EXISTS( SELECT \*

FROM P.work\_plan T

WHERE EXISTS( SELECT \*

FROM T.participating G

WHERE G.name='John'

)

OR T.leader.name='John' );

b) SELECT Documents: ( SELECT d.name FROM P.document d )

FROM Projects P

WHERE EXISTS ( SELECT \*

FROM P.work\_plan T

WHERE T.description\_task='Database design' );

c) SELECT UNIQUE ProjectName: P.name

FROM Projects P, P.work\_plan T

WHERE COUNT( SELECT \*

FROM Articles A, A.author AA

WHERE T.leader.name=AA.name ) >= 10;

d) SELECT UNIQUE ProjectName: P.name

FROM Projects P, P.work\_plan T

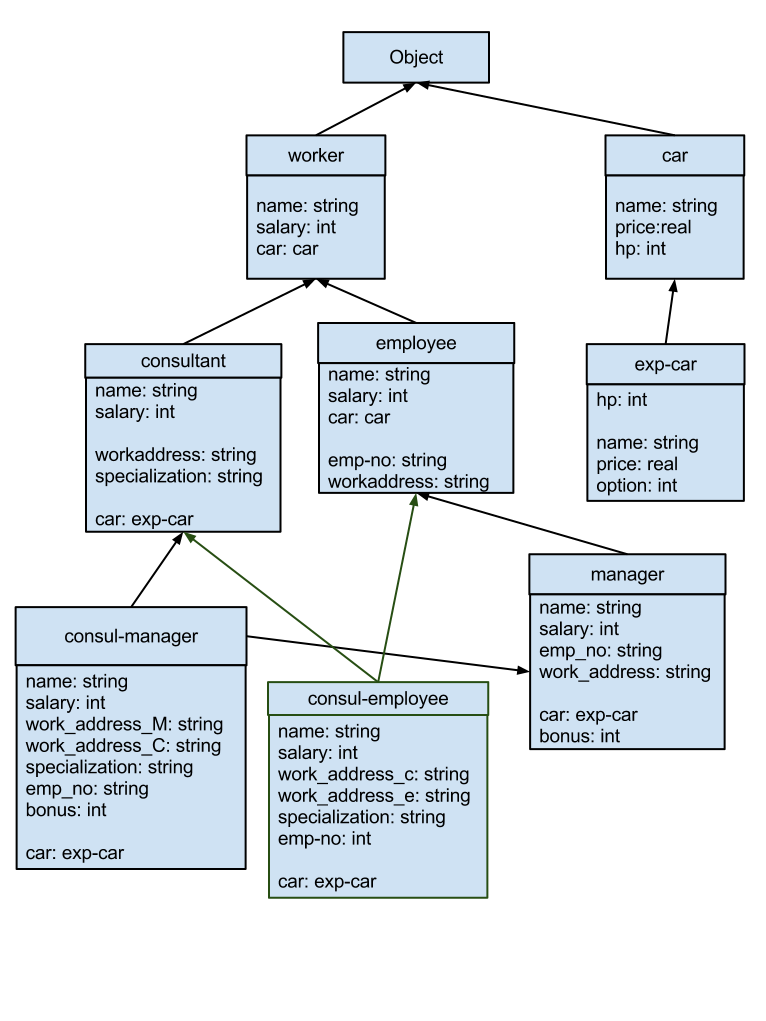
WHERE EXISTS( SELECT \*

FROM Articles A, A.author AA

WHERE T.leader.name=AA.name );

4)

a)



b)

i) Consul-Employee inherits:

- name: string (worker)

- salary: int (employee)

- work\_address\_c: string (consultant)

- work\_address\_e: string (employee)

- specialization: string (consultant)

- emp-no: int (employee)

- car: exp-car (redefined in consultant based on type assumption)

ii)

- Distinct name invariant

- Single origin invariant

- Complete inheritance invariant

- Domain compatibility invariant (due to consultant redefinition)

- Multiple inheritance rules R1, R2 and R3

- Class hierarchy manipulation rule R10