1.a Develop an assertion, written in SQL, that enforces the requirement that every CMIS relationship object has a source object that has a type that is allowed by the relationship object type of the relationship object.

Answer:

Create assertion rel\_relobjsource check

Not exists (

Select r.id

from relationship r,

cmis\_object c,

primary\_object\_type p,

secondary\_object\_type s,

object\_type o,

relationship\_object\_type ro

where r.id=c.id

and (c.id=p.id or c.id=s.id)

and (p.id=o.id or s.id=o.id)

and (o.id=ro.id));

create assertion RelationshipConsistency check

not exists (

select \*

from Relationship r, CMISObject relationshipObject, RelationshipObjectType rot

where r.id = relationshipObject.id

and relationshipObject.objectType = rot.id

and not exists (

select \*

from CMISObject sourceObject, RelationshipObjectTypeAllowedSourceTypes allowed

where sourceObject.id = r.source

and rot.id = allowed.relationshipObjectType

and allowed.allowedSourceType=sourceObject.objectType)

and not exists(

select \*

from CMISObjectSecondaryObjectType, RelationshipObjectTypeAllowedSourceTypessourceSecondaryType,

where sourceSecondaryType.isSecondaryTypeOf=r.source

and rot.id = allowed.relationshipObjectType

and allowed.allowedSourceType=sourceSecondaryType.secondaryObjectType

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1.b Give the user 'kong.f@husky.neu.edu' read-only access to all queryable documents whose query name begins with "kong".

Answer:

Create view kongview as

Select \*

From DocumentObjectType d

Where d.queryable = true

And d.queryName like ‘kong%’ ;

Grant select on kongview to 'kong.f@husky.neu.edu';

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. The following are the main commands being performed on this database:

* Given an employee specified by name, find the machines that the employee can repair.
* Find a product type given the first part of its title.
* List all products, showing all of its attributes as well as the code of its machine, that have yet to be completed.
* Modify the qualification level of an employee, given by name, for a product type given by title.
* Find a factory given its address.

Design the indexes for the database.

Answer:

Here is the recommended index design:

**Factory** table: hash index on the primary key, hash index on address

**ProductType** table: hash index on the primary key, btree index on the title.

**Employee** table: hash index on the primary key, hash index on name

**Product** table: hash index on the primary key, hash index on type, hash index on processedBy, hash index on completed

**Machine** table: hash index on the primary key.

**Qualification table:** hash index on the primary key.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_