DBS Assignment III

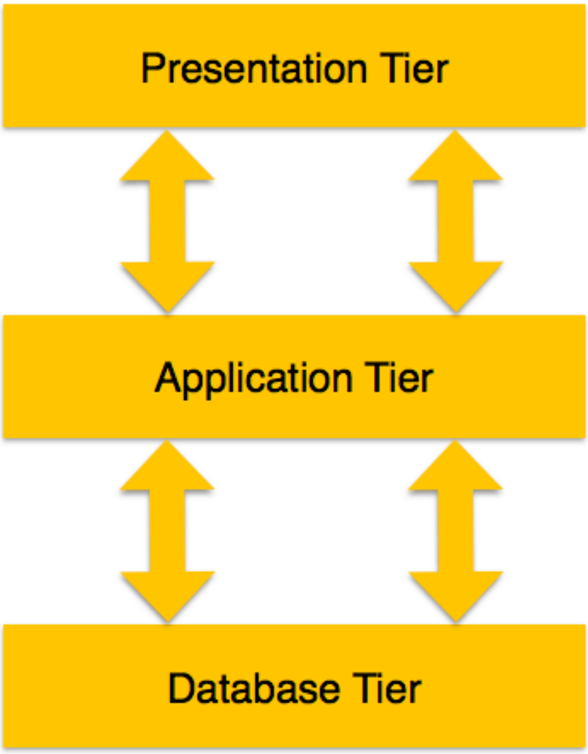
1. **List the major components of a DBMS architecture and describe their functions.**

The design of a DBMS depends on its architecture. It can be centralized or decentralized or hierarchical.

In 1-tier architecture, the DBMS is the only entity where the user directly sits on the DBMS and uses it.

If the architecture of DBMS is 2-tier, then it must have an application through which the DBMS can be accessed.

A 3-tier architecture separates its tiers from each other based on the complexity of the users and how they use the data present in the database. It is the most widely used architecture to design a DBMS.



Database tier - At this tier, the database resides along with its query processing languages. We also have the relations that define the data and their constraints at this level.

Application tier - At this tier reside the application server and the programs that access the database. For a user, this application tier presents an abstracted view of the database. End-users are unaware of any existence of the database beyond the application. At the other end, the database tier is not aware of any other user beyond the application tier.

User/Presentation tier - End-users operate on this tier and they know nothing about any existence of the database beyond this layer. At this layer, multiple views of the database can be provided by the application. All views are generated by applications that reside in the application tier.

2. **Consider the relational database as given below and give an expression in the relational algebra to express each of the following queries:**

```
employee(person_name, street,city)
works(person_name, company_name, salary)
company(company_name,city)
manages(person_name, manager_name)
```

a. **Find the names of all employees who live in the same city and on the same street as do their managers.**

$\pi_{\text{person\_name}}$ ((employee $\bowtie$ manages) $\bowtie$ ($\sigma_{\text{manager-name = empngr.person-name} \wedge \text{employee.street = empmgr.street} \wedge \text{employee.city = empmgr.city}}$ (ρ_{empmgr} (employee))))

b. **Find the names of all employees who do not work for "First Bank Corporation" assuming each person works for exactly one company.**

$\sigma_{\text{company\_name} \neq \text{'First Bank Corporation'}}$ (works))

c. **Find the names of all employees who do not work for "First Bank Corporation" assuming people may not work for any company.**

$\pi_{\text{person\_name}}$ ((works) - ($\sigma_{\text{company\_name = 'First Bank Corporation'}}$ (works)))

d. **Find the name of all employees who earn more than every employee of "Small Bank Corporation".**

maxsbc $\leftarrow$ $\pi_{\text{max(salary)}}$ ($\sigma_{\text{company\_name = 'Small Bank Corporation'}}$ (works))

$\pi_{\text{employee.person\_name}}$ ($\sigma_{\text{salary > maxsbc}}$ (employee $\bowtie$ works))

3. **Consider the relational database as given below and give an expression in the relational algebra to express each of the following queries:**

```
Flights(flno, from, to, distance, departs)
Aircraft(aid, aname, range)
Certified(eid, aid)
Employees(eid, ename, salary).
```

Pilots are those employees who are certified on at least one aircraft. An aircraft can be used for any flight provided it has sufficient range. Pilots can pilot any flight provided they are certified on an aircraft with sufficient range.

a. **Find flno of flights that can be piloted by every pilot whose salary is over \$100,000. (Using Division Operator.)**

$$R_1 \leftarrow \pi_{\text{eid}} (\sigma_{\text{salary} > 100000} (\text{Employees} \bowtie \text{Certified}))$$
$$R_2 \leftarrow \pi_{\text{eid, aid}} (\text{Employees} \bowtie \text{Certified}))$$
$$\text{aids} \leftarrow R_2 \div R_1$$

$$\pi_{\text{flno}} (\sigma_{\text{distance} \leq \text{range}} (\text{Flights, aids}))$$

b. **Find eid's of pilots who are certified on some Boeing.**

$$\pi_{\text{eid}} (\text{Certified} \bowtie (\sigma_{\text{aname LIKE '\%Boeing\%'}} (\text{Aircraft})))$$

c. **Find names of pilots who can operate planes with a range greater than 3,000 miles, but are not certified on any Boeing.**

$$\pi_{\text{ename}} (\text{Certified} \bowtie \text{Employees} \bowtie (\sigma_{\text{range} > 3000 \wedge \text{aname NOT LIKE '\%Boeing\%'}} (\text{Aircraft}))$$

d. **Find aid's of aircraft that can fly non-stop from LA to NY. Assume you don't already know the distance.**

$$\pi_{\text{aid}} (\sigma_{\text{range} \geq \text{distance}} (\text{Aircraft x} (\pi_{\text{distance}} (\sigma_{\text{from = 'LA'} \wedge \text{to = 'NY'}} (\text{Flights})))))$$

4. What are the different design issues while designing an ER Diagram? Explain how a non binary relationship can be converted into a binary relationship?

It is not always easy to decide on the best way to model the reality one wishes to describe.

- The design process is an iterative process.
- You start with an initial design and gradually refine that.

Sometimes it is hard to decide whether something should be represented as an entity, an attribute or a relationship.

- There are often several ways to model the reality you want to describe.
- You cannot always say that one way is right or wrong; some of these decisions are simply design matters.

How you design your model is very much dependent on the company or the organization that you model, and the role of the things modeled have in the functioning of the organization.

- Things crucial to the organization should get a more eminent role in the model (e.g. rather an entity than merely an attribute)

Some relationships that appear to be nonbinary could actually be better represented by several binary relationships.

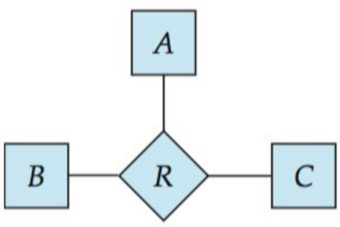
It is always possible to replace a non-binary (n-ary, for $n > 2$) relation- ship set by a number of distinct binary relationship sets. For simplicity, consider the abstract ternary ($n = 3$) relationship set R, relating entity sets A, B, and C. We replace the relationship set R by an entity set E , and create three relationship sets as:

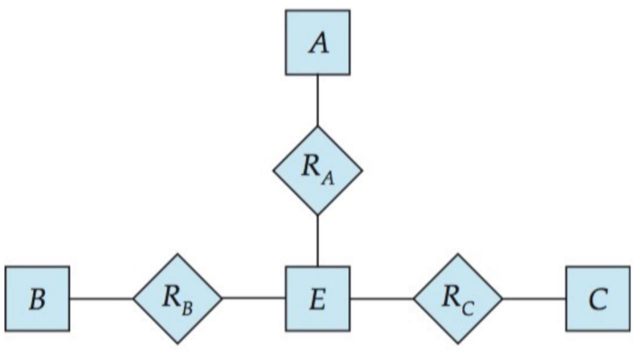
R_A relating E and A

R_B,relating E and B

R_C relating E and C

In simple terms, For each relationship a_i in the relationship set R, we create a new entity e_i in the entity set E. Then, in each of the new relationship sets, we insert a relationship (a_i, e_i) in R.





5. **Explain the design constraints on specialization and generalization with proper suitable examples.**

Specialization is opposite to Generalization. It is a top-down approach in which one higher level entity can be broken down into two lower level entity. In specialization, some higher level entities may not have lower-level entity sets at all.

Generalization is a bottom-up approach in which two lower level entities combine to form a higher level entity. In generalization, the higher level entity can also combine with other lower level entity to make further higher level entity.

Constraint on which entities can be members of a given lower-level entity set.

- **condition-defined**, for eg, all customers over 65 years are members of senior-citizen entity set; senior-citizen ISA person.
- **user-defined.**

Constraint on whether or not entities may belong to more than one lower- level entity set within a single generalization.

- **Disjoint** - an entity can belong to only one lower-level entity set, noted in E-R diagram by having multiple lower-level entity sets link to the same triangle.
- **Overlapping** - an entity can belong to more than one lower-level entity set.

Completeness constraint - specifies whether or not an entity in the higher-level entity set must belong to at least one of the lower- level entity sets within a generalization.

- **Total** - an entity must belong to one of the lower-level entity sets.
- **Partial** - an entity need not belong to one of the lower-level entity sets.