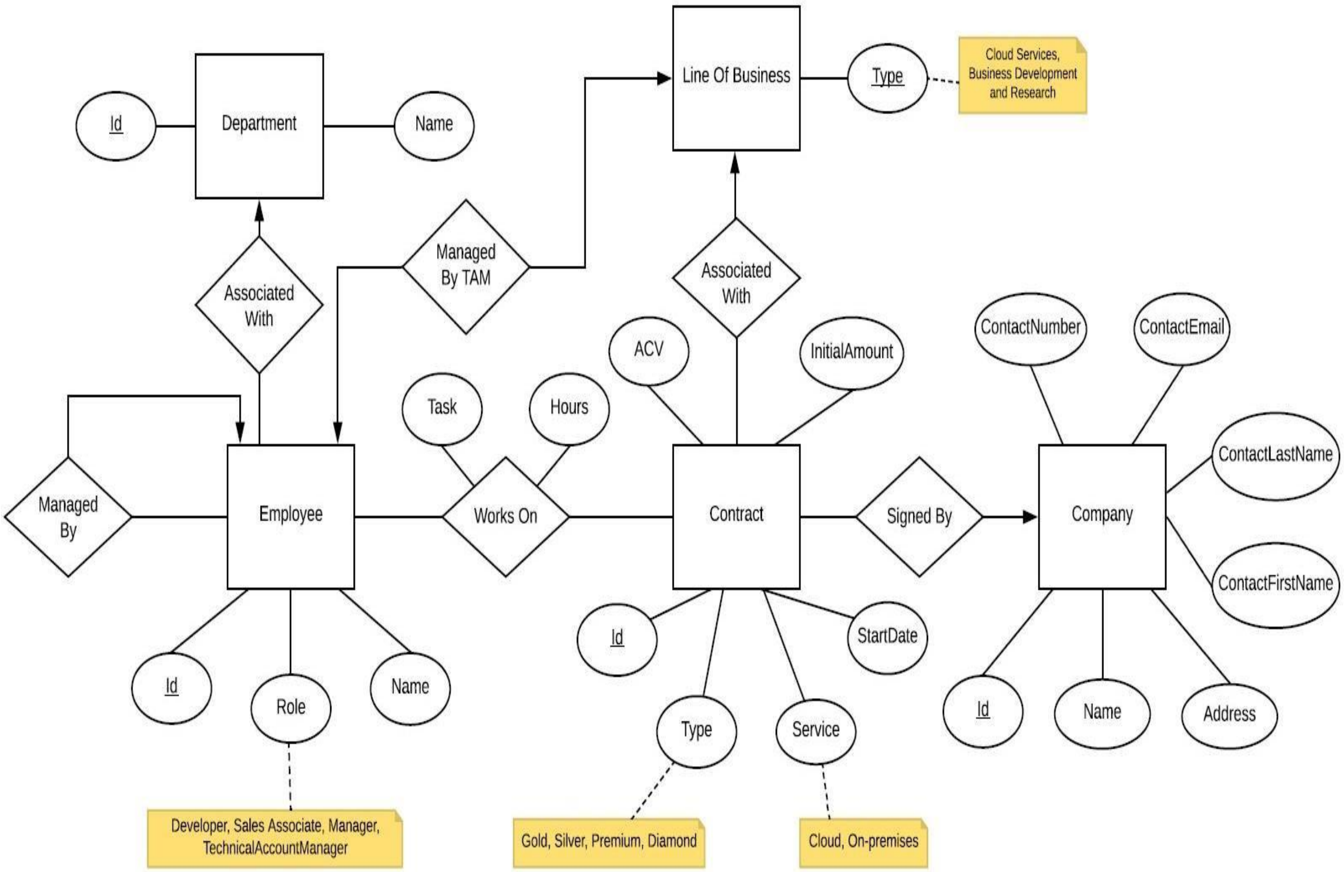
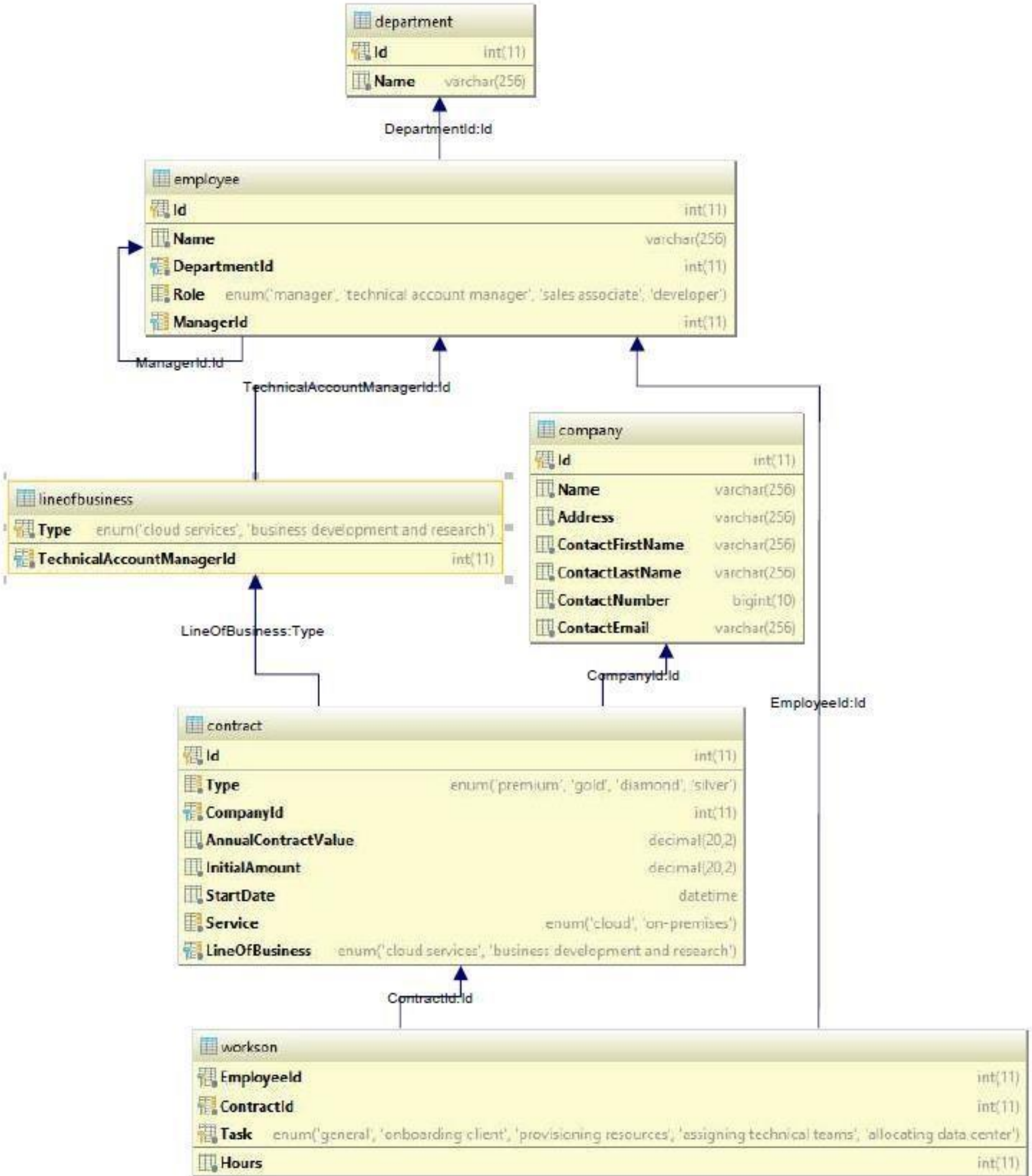
COMP -353

Assignment#2

Sample solution



ER Diagram for Contract Management System



Tables and data types

Question 1

A manager allocates different type of tasks (as per req.1) to all the employees. Using this information about the allocated tasks, an employee logs in the number of hours worked on a specific task. For this requirement design your database, give a list of all employees who are working on contracts with ACV of $50,000 and working on “Provisioning of resources" task.

The database schema was updated to support enumerated task types and hours logged.

The following query was used to find all employees working on gold contracts and the “Onboarding of Client” task. With the help of a join between the relevant tables and searches for contracts of type gold where the employee’s task required is **“Provisioning of resources”.**

Listing 1 – SQL query for question 1

```
SELECT E.Id, E.Name, E.Role
FROM Employee E
  INNER JOIN WorksOn WO ON WO.EmployeeId = E.Id
  INNER JOIN Contract C on WO.ContractId = C.Id
WHERE C.Type = 'Gold' AND WO.Task = ‘Provisioning of resources’ ORDER BY E.Id;
```

Question 2

Give the list of all the employees who are not managers working in “Cloud services” line of business.

```
SELECT E.Id, E.Name, E.Role
FROM Employee E
  INNER JOIN WorksOn WO ON WO.EmployeeId = E.Id
  INNER JOIN Contract C on WO.ContractId = C.Id
WHERE E.Role != 'Manager' AND E.Role != 'Technical Account Manager' AND
C.LineOfBusiness = 'Cloud Services'
ORDER BY E.Id;
```

A technical account manager oversees managers working on contracts in that line of business.

Question 3

Find the canonical cover of the following:

$\boldsymbol{R} = (\boldsymbol{D}, \boldsymbol{E}, \boldsymbol{G})$

$\boldsymbol{F} = \{\boldsymbol{D} \rightarrow \boldsymbol{EG}, \boldsymbol{E} \rightarrow \boldsymbol{G}, \boldsymbol{D} \rightarrow \boldsymbol{E}, \boldsymbol{DE} \rightarrow \boldsymbol{G}\}$

To find the canonical cover, find the set of functional dependencies $G = F$ that is minimal. We do this by successive reductions of F using the following steps:

Step 1: Put all FDs in simple form where RHS has a single

attribute Step 2: For all FDs, check if LHS can be reduced:

Check $(G - \{AB \rightarrow C\} \cup \{A \rightarrow C\})^+ = F^+$then can simply use $A \rightarrow C$ or $(G - \{AB \rightarrow C\} \cup \{B \rightarrow C\})^+ = F^+$ then can simply use $B \rightarrow C$

Step 3: For all FDs $X \rightarrow A$ check if A can be obtained from X^+ without $X \rightarrow A$:

$$(G - \{X \rightarrow A\})^+ = F^+$$

Then $X \rightarrow A$ is redundant.

Step 1:

We write G in simplified form such that there is only 1 attribute on the RHS of all FDs. In this form we see $D \rightarrow E$ is repeated twice so it can be removed. After step 1 the cover is:

$$G = \{D \rightarrow E, D \rightarrow G, E \rightarrow G, DE \rightarrow G\}$$

Step 2:

The only FD that can be left reduced is $DE \rightarrow G$. We check if either D^+ or E^+ can obtain G.

$$D^+ = \{D, E, G\}$$

$$E^+ = \{E, G\}$$

Since $G \in D^+$ and $G \in E^+$ we can replace $DE \rightarrow G$ with $D \rightarrow G$ and $E \rightarrow G$. However, both of these already exist so $DE \rightarrow G$ is entirely redundant. After step 2 the cover is:

$$G = \{D \rightarrow E, D \rightarrow G, E \rightarrow G\}$$

Step 3:

For each FD we check if the RHS of the FD can be obtained from the cover while excluding the FD.

$\underline{D \rightarrow E}$:

Exclude $D \rightarrow E$ and calculate D^+:

$D^+ = \{D, G\}$

$E \notin D^+$ so $D \rightarrow E$ is not redundant $\underline{D \rightarrow G}$:

Exclude $D \rightarrow G$ and calculate D^+

$D^+ = \{D, E, G\}$

$G \in D^+$ so G can be obtained from D without the need of $D \rightarrow G$. Thus $D \rightarrow G$ is redundant.

$\underline{E \rightarrow G}$:

Exclude $E \rightarrow G$ and calculate D^+

$E^+ = \{E\}$

$G \notin E^+$ so $E \rightarrow G$ is not redundant

After all minimizations the canonical cover becomes:

$$\boxed{G = \{D \rightarrow E, E \rightarrow G\}}$$

<u>Question 4</u> Given:

$$\boldsymbol{R(employeefirstname,\ address,\ gender,\ employeerole, employeesalary)}$$

with the following functional dependencies that hold on R:

$\boldsymbol{FD1: employeeFirstName \rightarrow address, gender}$
$\boldsymbol{FD2: address \rightarrow employeeRole,}$ ***seniority***
$\boldsymbol{FD3: employeeRole, gender \rightarrow employeeSalary}$

a) Find all candidate keys of R:

A candidate key must be an attribute or combination of attributes from which we can obtain all other attributes in R and must also be minimal. We can compute the cover of each attribute to check this:

$employeeFirstName^{+}$ = $\{employeeFirstName,\ address,\ gender,$ ***seniority*** $, employeeRole,\ employeeSalary\}$

$address^{+}$ = $\{address,\ employeeRole,$ ***seniority*** $\}$

$$employeeRole^{+} = \{employeeRole\}$$

$$gender^{+} = \{gender\}$$

$$salary^{+} = \{salary\}$$

$$(employeeRole,\ gender)^{+} = \{employeeRole,\ gender,\ employeeSalary\}$$

We see that $employeeFirstName$ can be used to obtain all other attributes and is minimal hence it is a candidate key. Also note that any attribute which does not appear on the RHS of a functional dependency should appear in the candidate key, which only applies to employeeFirstName in this case. We can obtain other super keys by augmenting other attributes with $employeeFirstName$ however they will not be minimal and are thus not candidate keys.

$$\boxed{candidate\ key = employeeFirstName}$$

b) Find a 3NF decomposition of R that is lossless and dependency preserving.

To be lossless, it is required that the attributes joined between two decompositions forms a key of one

To be dependency preserving, the union of the FDs from the decompositions must equal the original set F of functional dependencies. To be in 3NF each FD $X \rightarrow A$ must satisfy <u>at least one</u> of the following:

- $X \rightarrow A$ is a trivial FD
- X is a key
- A is part of a key

We begin by inspecting each functional dependency in R to see if any violate the 3NF requirement. Please add seniority where needed but it won't have any impact on the final answer.

$employeeRole, gender \rightarrow employeeSalary$ violates 3NF because employeeRole, gender is not a key $\in R$. So, the relationship must be further subdivided into R_1 and R_2. In R_2 $address \rightarrow employeeRole$ also violates 3NF because address is not a key $\in R_2$. Thus R_2 is divided once again into R_3 and R_4 (see Figure 1 for visual reference).

Once decomposed into R_1, R_3, R_4 all functional dependencies meet the 3NF requirements. We can see that the decomposition is also lossless because the intersection when merging tables is a key of at least one of the two tables (note that R_3 and R_4 are to be joined first into R_2 before joining with R_1)

It is also dependency preserving since the union of all functional dependencies made from R_1, R_3 and R_4 forms the original set of functional dependencies.

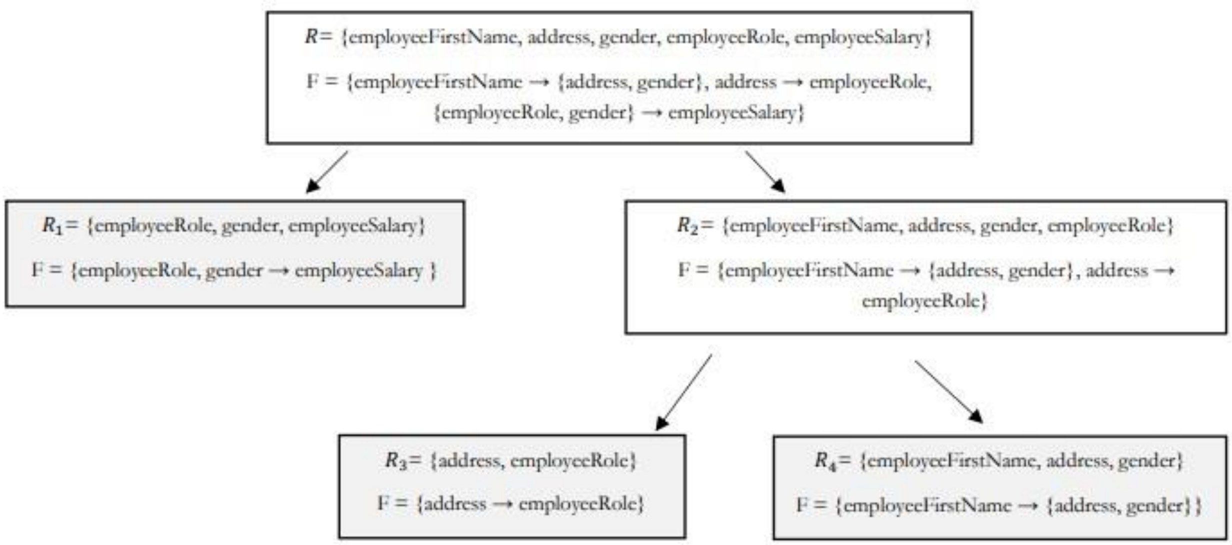


Figure 1 – Decomposition process for 3NF

The final decomposed relationships are:

$$\boxed{R_1(\underline{employeeRole, gender}, employeeSalary)}$$

$$\boxed{R_3(\underline{address}, employeeRole)}$$

$$\boxed{R_4(\underline{employeeFirstName}, address, gender)}$$

c) Identify all primary keys of the decomposed 3NF relations.

The following primary keys result from the three decompositions:

$$\boxed{R_{1,pk} = employeeRole, gender}$$

$$\boxed{R_{3,pk} = address}$$

$$\boxed{R_{4,pk} = employeeFirstName}$$