This section contains a transcript of an actual database design session using VMITS. The session has been divided into sections to make it more readable.

A. Initialization

User? Please enter your name followed by a dot (.) and return. If more than one word is needed, connect them by using underscore (_) character. Always respond in this way and use only small letters. The VMITS prompt for the user is "-->".

Now enter your name.

--> tripathy.

Hello tripathy.

For what purpose do you want to develop a database?

--> personnel_and_project.

Enter the name of first user view.

--> employee_view.

A.1. Entities

What are the things of interest to you in the employee_view?

I shall refer to them as entities.

--> employee.
What are the attributes (properties that describe the entity) for "employee"?

--> emp_id name address post salary.

Any more attributes for "employee"?

("see" / "yes" / "no")

--> no.

Fine.

Here are the attributes for

employee : [ emp_id, name, address, post, salary ]

Can a single "employee" have more than one value of these attributes?

("no" / "why ?" / attribute(s) which can have more than one value / "end" if you do not know ).

--> address

I will delete the attribute "address" and this will be added as an entity later.

A relationship should be defined between "employee" and "address".

Here are the attributes for

employee : [ emp_id, name, post, salary ]

Enter attributes which can identify uniquely an instance of entity "employee".
now let us deal with "address".

Is the term "address" in singular form?

("Yes" / "no")

--> Yes.

Enter the attributes for "address".

--> city street pin_code.

Any more attributes for "address"?

("see" / "yes" / "no")

--> no.

Can a single "address" have more than one value for any of these attributes?

("no" / "why ?" / attribute(s) which can have more than one value / "end" if you do not know).

--> no.

Here are attributes for

address : [ city, street, pin_code ]

Enter attributes which can identify uniquely an instance of "address".
Enter the name of the next entity / "end".

Enter the attributes for "manager".

Any more attributes for "manager"?

("see" / "yes" / "no")

-- > no.

Here are the attributes for

manager : [ emp_id, name, title, salary ]

Can a single "manager" have more than one value for any of these attributes?

("no" / "why?" / attribute(s) which can have more than one value / "end" if you do not know).

-- > no.

Here are attributes for

manager : [ emp_id, name, title, salary ]

Enter attributes which can identify uniquely an instance of manager.

-- > emp_id.
Fine.

Enter the name of next entity / "end".
--> department.

Enter the attributes for "department".
--> dept_no name.

Any more attributes for "department"?
("yes" / "no").
--> no.

Fine.

Here are the attributes for department: [ dept_no, name ]

Can a single "department" have more than one value for any of these attributes?
("no" / "why ?" / attribute(s) which can have more than one value / "end" if you do not know)
--> no.

Fine.

Here are the attributes for department: [ dept_no, name ]

Enter attributes which can uniquely identify an instance of department.
--> dept_no.
Fine.
Enter the next entity / "end".

--> division.

Enter the attributes for "division".

--> divn_no.

Any more attributes for "division"?

("yes" / "no").

--> no.

Fine.

Here are attributes for division: [ divn_no ]

Enter attribute(s) which can uniquely identify an instance of "divn_no".

--> divn_no.

Fine.
Enter next entity / "see" / "end".

--> end.

Here are the entities with their attributes and identifiers of the "employee_view"

employee: [ EMP_ID, name, post, salary ]

address: [ STREET, pin_code, city ]

manager: [ EMP_ID, name, title, salary ]

division: [ DIVN_NO ]
End entities.
Do you wish to add, delete or change any of these entities? ("no" / "add" / "delete" / "change").
--> no.
Fine.

A.2. ID Relationships
Enter entity which depends on other entity for its identification / "see" / "end".
--> department.
On which entity the entity "department" depends for its identification?
--> division.
Fine.
Enter next entity which depends on other entity for its identification / "see" / "end".
--> end.
Fine.

A.3. IS_A Relationships
I need to know if an entity is a specialization of another entity, e. g. STUDENT_VIEW

GRADUATE_STUDENT IS_A STUDENT.

Please enter your information with three words "A IS_A B" on
each line followed by a dot and Return.
First, would you like to see the entities again?
("yes" / "no").
--> no.
Fine.
Enter IS_A relationship of the form of "A IS_A B".
--> manager IS_A employee.
Enter next IS_A relationship / "end".
-> end.
Fine.

A.4. Relationship
You have described the things of "employee" view of the database. I need to know how these things interact with each other.
I want to know the association between pair of entities and the association between three entities.
Enter the binary relationship in the form of "A Verb_Phrase B" and ternary relationship in the form of "A VP B PR C" / "see" / "see_ent" / "end".
--> employee works_in department.
Is there at most one "employee" for every "department"?
("yes" / "no").
--> no.
Is there at most one "department" for every "employee"?
("yes" / "no").
--> yes.

An attribute of a relationship is a property or characteristic of the relationship.
For example, Relationship "Student Attends Course"

   Acceptable attribute : "Grade".

Enter attributes for "employee works_in department" / "end".
--> location.

Any more attribute for relationship:
"employee works_in departmant"?
("yes" / "no").
--> no.

Fine.

Enter next relationship / "see" / "see_ent" / "end".
--> end.

A.5. Missing relationships

A relationship is needed between the entity "employee" and the entity "address".

Please enter a relationship between "employee" and "address" / "why ?".
--> why ?
When you defined the entity "employee" originally, you said that "address" is one of the attributes for "employee". This implies that a relationship should exist between these two entities.

Please enter a relationship between "employee" and "address".

--> employee located_in address.

Is there at most one "employee" for every "address"?

("yes" / "no").

--> no.

Is there at most one "address" for every "employee"?

--> no.

Enter attributes for "employee located_in address" / "end".

--> end.

Fine.

Here is your set of entities and relationships:

employee : [ EMP_ID, name, post, salary ]

address : [ STREET, PIN_CODE, city ]

manager : [ EMP_ID, name, title, salary ]

department : [ DEPT_NO, name ]

division : [ DIVN_NO ]

End entities.

department ID division.

manager IS_A employee.

employee works_in department [location].
employee located in address.

End relationships.

Do you want to modify any of entities?
("see" / "yes" / "no").
--> no.

Do you want to modify any of relationships?
("see" / "yes" / "no").
--> no.

Fine.

A.6. Adjust attributes

"department ID division".

The identifier of "division" is added to the identifier of "department".

department : [ DEPT_NO, DIVN_NO, name ]

"manager IS_A employee".

The identifier of "manager" is made same as that of "employee" and all the non-key attributes of "manager" which also belong to "employee" are deleted.

employee : [ EMP_ID, name, post, salary ]

manager : [ EMP_ID, title ].

The system will add the entity name as a prefix to common attributes and entities.
employee: [ EMP_ID, employee_name, post, salary ].
manager: [ EMP_ID, title ].
The system will add the entity name as a prefix to common attributes of entities.
employee: [ EMP_ID, employee_name, post, salary ].
department: [ DEPT_NO, DIVN_NO, department_name ].

B. Next View Creation

Is there any other view of the database to be created?
("yes" / "no" / "see").
--> yes.
Enter the name of the next view to be created.
--> project_view.

B.1. Entities

Enter the name of entity in "project" view of the database.
--> employee.
What are the attributes for "employee"?
--> emp_id name qualification salary.
Any more attributes for "employee"?
("see" / "yes" / "no").
--> no.
Fine.

208
Here are the attributes for:
employee: [ emp_id, name, qualification, salary ]
Can a single "employee" have more than one value for any of these attributes?
("no" / "why" / attribute(s) which can have more than one value for any of these attributes / "end" if you do not know).
--> end.
Fine.
Enter attributes which can identify uniquely an instance of "employee".
--> emp_id.
Fine.
Enter next entity / "end".
--> project.
Enter attributes for "project".
--> proj_no name.
Any more attributes for "project"?
("see" / "yes" / "no").
--> no.
Fine.
Here are attributes for
project: [ proj_no, name ].
Can a single "project" have more than one value for any of
these attributes?
("no" / "why" /attribute(s) which can have more than one value for any of these attributes / "end" if you do not know).
--> no.
Fine.
Enter attributes which can identify uniquely an instance of "project" / "end".
--> proj_no.
Fine.
Enter next entity / "end".
--> skill.
Enter attributes for "skill".
--> skill_no.
Any more attributes for "skill"?
("see" / "yes" / "no").
--> no.
Enter the attributes which can identify uniquely an instance of "skill".
--> skill_no.
Fine.
Enter next entity / "end".
--> end.
Here are entities with their attributes and identifiers in "project_view"
employee: [EMP_ID, name, qualification, salary]
project: [PROJ_NO, name]
skill: [SKILL_NO]

B.2. ID Relationships

Enter entity name which depends on other entity for its identification / "see" / "end".
--> end.
Fine.

B.3. IS_A Relationship

Enter IS_A relationship in the form of "A IS_A B" / "end".
--> end.

B.4. Relationships

Enter binary relationship in the form of "A VP B" and ternary relationship in the form of "A VP B PR C" / "see" / "see_ent" / "end".
--> employee use skill for project.

Is there at most one "employee" for every "skill" and "project"?
("yes" / "no").
--> no.
Is there at most one "skill" for every "employee" and "project"?
("yes" / "no").
--> no.
Is there at most one "project" for every "employee" and "skill"?
("yes" / "no").
--> no.
Enter next relationship / "see" / "See_ent" / "end".
--> end.
Here is your set of entities and relationships:
employee : [ EMP_ID, name, qualification, salary ]
project : [ PROJ_NO, name ]
skill : [ SKILL_NO ]
End Entities.
employee use skill for project.
End relationships.
Do you wish to modify any of the entities?
("see" / "yes" / "no").
--> no.
Do you wish to modify any of the relationships?
("see" / "yes" / "no").
--> no.
Fine.

B.5. Missing Relationships
/* there are no missing relationships identified */

B.6. Adjust Attributes

employee : [ EMP_ID, name, qualification, salary ]
project : [ PROJ_NO, name ]

The system will add the entity name as a prefix to common attributes of entities.

employee : [ EMP_ID, employee_name, qualification, salary ]
project : [ PROJ_NO, project_name ]

C. View integration

The system will integrate the first view with the next view.

C.1. Entity integration

Here are the set of entities along with attributes and identifiers.

employee : [ EMP_ID, employee_name, post, qualification, salary ]
address : [ STREET, PIN_CODE, city ]
manager : [ EMP_ID, title ]
department: [ DEPT_NO, DIVN_NO, department_name ]
division: [ DIVN_NO ]
project: [ PROJ_NO, project_name ]
skill: [ SKILL_NO ]

C.2. Relationship integration

Here are set of relationships of the integrated view:
department ID division
manager IS_A employee
employee works_in department [location]
employee located_in address
employee use skill For project.

C.3. Next view creation

Is there any more view to be created?
("yes" / "no" / "see").
--> no.
Fine.

D. Conceptual view design

Here is the set of entities and relationships of the conceptual view.
employee: [EMP_ID, employee_name, post, qualification, salary]
address : [ STREET, PIN_CODE, city ]
manager : [ EMP_ID, title ]
department : -[ DEPT_NO, DIVN_NO, department_name ]
division : [ DIVN_NO ]
project : [ PROJ_NO, project_name ]
skill : [ SKILL_NO ]

end entities.

employee works_in department [location]
employee located_in address
employee use skill For project.

end relationships.

E. Logical view design

E.1. Representation phase

Here are the relation schemas alongwith attributes and keys :

employee : [ EMP_ID, employee_name, post, qualification, salary ]

address : [ STREET, PIN_CODE, city ]
manager : [ EMP_ID, title ]
department : [ DEPT_NO, DIVN_NO, department_name ]
works_in : [ EMP_ID, DEPT_NO, DIVN_NO, location ]
use_for : [ EMP_ID, PROJ_NO, SKILL_NO ]
located_in : [ STREET, PIN_CODE, EMP_ID ]

Relation Schemas with only one attribute are deleted.

215
E.2. Normalization

/* Second normal Form */
address : [ STREET, PIN_CODE, city ]
Which of the following do you need before you know the "city"?
   a) STREET
   b) PIN_CODE
   c) STREET and PIN_CODE
"a" / "b" / "c" / "no" (if you do not know).
--> c.
Fine.
department : [ DEPT_NO, DIVN_NO, department_name ]
Which of the following do you need before you know the "department_name" ?
   a) DEPT_NO
   b) DIVN_NO
   c) DEPT_NO and DIVN_NO.
"a" / "b" / "c" / "no" (if you do not know).
--> c.
Fine.

/* Third normal Form */
For each of the following relation schema, identify any non-key attribute(s) (written in lower-case which depends
directly upon any other non-key attribute(s).
Respond with: "non-key(s) depend_on non-key(s)".
Separate the attributes by a blank.

employee : [ EMP_ID, employee_name, post, qualification, salary ]
Enter any non-key attribute(s) which depends on any other non-key attribute(s) / "end".
--> end.
Fine.

works_in : [ EMP_ID, DEPT_NO, DIVN_NO, location ]
Enter any non-key attribute(s) which depends on any other non-key attribute(s) / "end".
--> end.
Fine.

F. End Session

Here is the normalized relation schemas representing the database "Personnel_and_project" :

employee : [ EMP_ID, employee_name, post, qualification, salary ]
address : [ STREET, PIN_CODE, city ]
manager : [ EMP_ID, title ]
department : [ DEPT_NO, DIVN_NO, department_name ]
works_in : [ EMP_ID, DEPT_NO, DIVN_NO, location ]

use_for : [ EMP_ID, PROJ_NO, SKILL_NO ]

located_in : [ STREET, PIN_CODE, EMP_ID ]

218