Unit-4

Design

Compiled by Bikash Magar
bikashtm17@gmail.com

Database Design

- ***Database design can be categorized as logical and physical design.***
- **In logical design**

- It is a description of entities that are defined and how they are related to each other and what kind of data is to be stored.
- Logical database design involves creating a data model that shows an organization's information. For.Eg. Entity Relation Diagram(E-RD)
- Logical database design is a duty of a database administrator and involves gathering a business' organization and processes so the database can accommodate the business. After this data has been gathered, you could create a logical model to map the functions and relationships between the processes and data.
- Once a logical model is diagrammed, you can make a physical database model that implements the information gathered for the logical model.
- ***Logical design*** results in the blueprint of the new system.

2. Physical Design

- After completing the logical design of your database, you now move to the physical design. You and your colleagues need to make many decisions that affect the physical design, some of which are listed below.

- How to translate entities into physical tables?
- What attributes to use for columns of the physical tables?
- Which columns of the tables to define as primary key?
- What views to define on the tables?
- How to normalize the tables?
- How to resolve many-to-many relationships? etc.

- During physical design, you transform the entities into tables, the instances into rows, and the attributes into columns.
- The physical design of your database optimizes performance while ensuring data integrity by avoiding unnecessary data redundancies.
- **The physical design** relates to the actual input and output processes of the system. Here, the physical design follows the logical design.
- **Physical Database Design** deals with how the data will be stored in the database using suitable DBMS and this design is generally created by database administrators and developers.

Differences

The major differences between logical database design and physical database design are as follows –

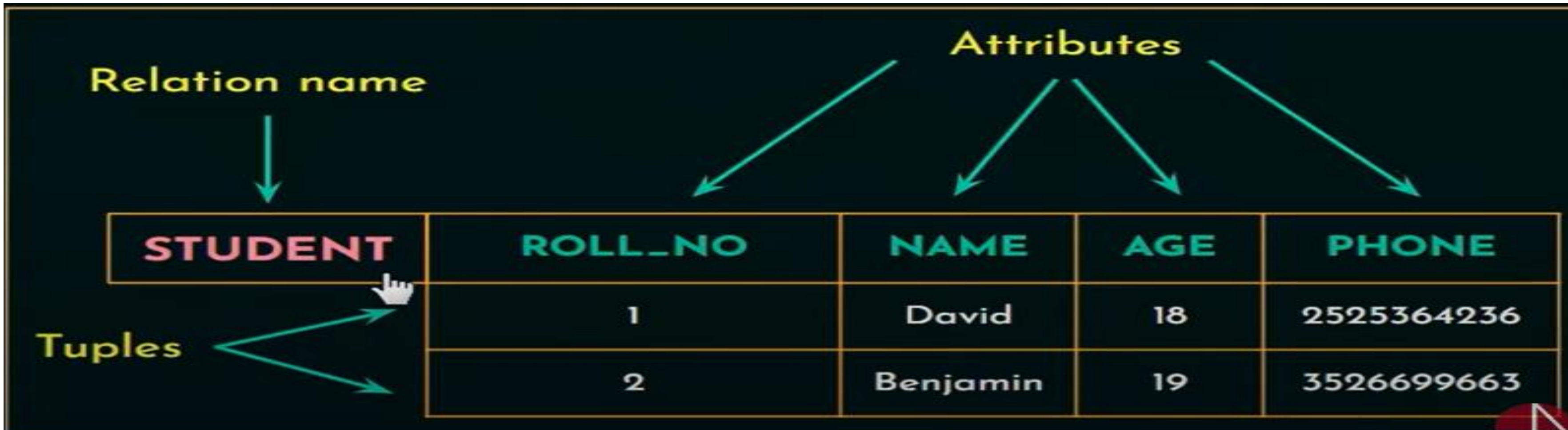
Logical Database Design	Physical Database Design
That describes the data without regard to how they will be physically implemented in the database.	That represents how the actual database is built.
Defines the data elements and their relationship.	Developing the actual database.
Simpler than the Physical database design.	Complex than the Logical database design.

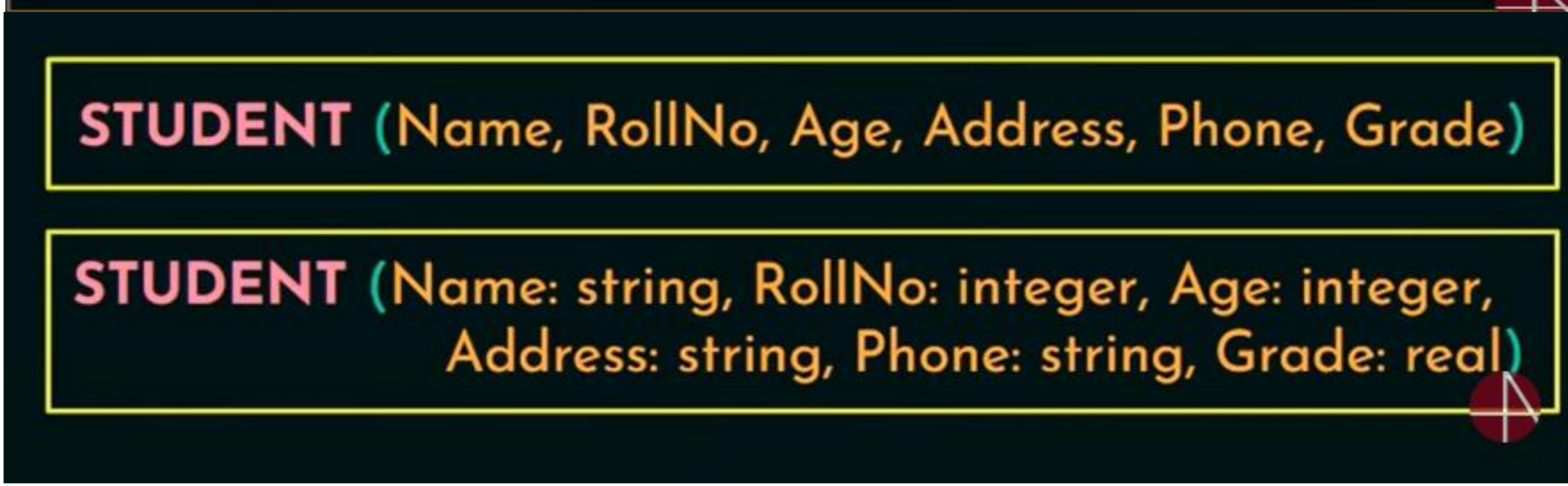
Relational Database Model

- To understand the structure of a database we need to know a few important terms. These are:
- **Relations** –Relational database represents data as a collection of tables. A table is also called a relation. It is a grid made up of rows and columns. Well in relational databases we often refer table as "relations".
- **Tuples** – A row in a table is known as a record, or a tuple. A tuple holds all the data on a single item. For example, if we have a table to store book details for a shop, then each tuple is an individual book the shop sells, and it will store all the data on that individual book such as the ISBN, book title and price.
- **Attributes** –Each column or the column header is called as attributes or fields. In the previous example we said that each record for a book will store the ISBN, book title and price. Well these three things are all examples of attributes. Attributes appear as columns in a database table.

• Relation Schema

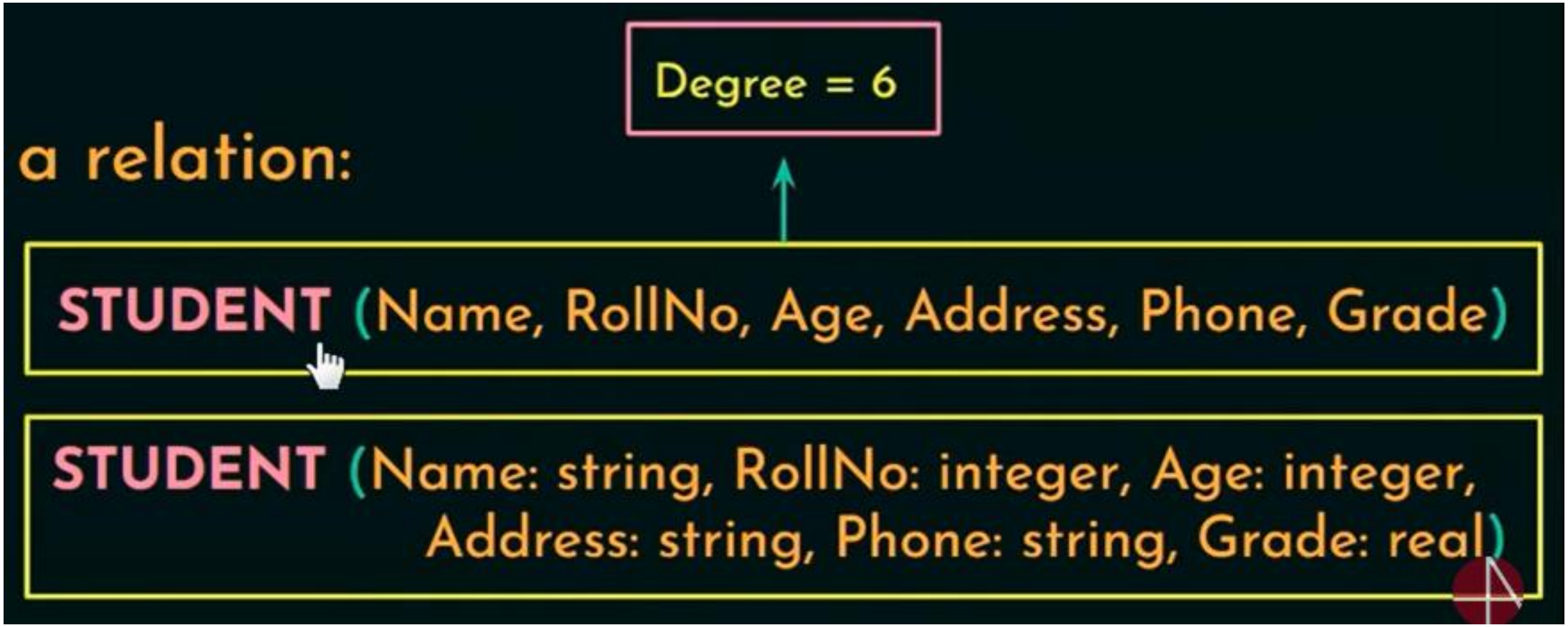
➢ Relational schema describes a relation made up of a relation name R and a list of attributes A1, A2, A3, A4, ……., An.





Degree(or arity) of an relation.

- It defined as the number of attributes in a relation schema.



Cardinality

- Cardinality is a total number of tuples present in a relation.

STUDENT	ROLL_NO	NAME	AGE
	1	Harry	19
Cardinality = 3	2	Ben	22
	3	Kathy	20

Relational State(or relation instance)

- It defines a set of tuples at any given time t.
- Lets say at time T1 the below given is the set of tuples at that given time.

STUDENT	ROLL_NO	NAME	AGE
Cardinality = 3	1	Harry	19
	2	Ben	22
	3	Kathy	20

- Suppose if I am updating relation by deleting the 3rd row then the set tuples say at time T2 would be only 1st and 2nd rows.

STUDENT	ROLL_NO	NAME	AGE
Cardinality	1	Harry	19
	2	Ben	22

- **Domains** – All the possible allowable values for an attribute.

For example, a field may have an integer number data type, which defines that it can only allow whole numbers to be entered. However, there may be additional rules applied, such as that the number must be between 1 & 10. The domain would therefore be this range of whole numbers.

Eg2. Name attribute that holds string of characters that represent name of persons.

Eg3. Age: integer type which contains Possible ages of employees of a company (values between 20 &70 years old.)

Normalization

- Normalization is a process of organizing the data in the database to remove or reduce redundancy or anomalies (insertion anomaly, update anomaly and deletion anomaly) from a table .

Anomalies in Database:

- A database anomaly is an inconsistency in the data resulting from an operation like an update, insertion, or deletion. There can be inconsistencies when a record is held in multiple places and not all of the copies are updated.
- Anomalies occur when the data present in the database has too much redundancy and if the tables making up the database are poorly constructed
- A database anomaly is a fault in a database.
- In most cases, this is removed through the normalization procedure, which involves the joining and splitting of tables.

- There are three types of anomalies that occur when the database is not normalized. These are ***Insertion, Updation, and deletion*** anomalies.
- Lets see an example to understand this,

Example

- Consider a manufacturing firm that keeps worker information in a table called employee, which has four columns: w_id for the employee's id, w_name for the employee's name, w_address for the employee's address, and w_dept for the employee's department. The table will look like this at some point:

w_id	**w_name**	**w_address**	**w_dept**
201	David	Delhi	F001
201	David	Delhi	F002
223	Mike	Agra	F890
266	Berry	Chennai	F900
266	Berry	Chennai	F004

Types of Anomalies in DBMS

- Various types of anomalies can occur in a DB. But these can be easily identified and fixed. The following are actually the ones about which we should be worried:

1.Update Anomaly

- Employee David has two rows in the table given above since he works in two different departments. If we want to change David's address, we must do so in two rows, else the data would become inconsistent.
- If the proper address is updated in one of the departments but not in another, David will have two different addresses in the database, which is incorrect and leads to inconsistent data.

2.Insert Anomaly

- If a new worker joins the firm and is currently *unassigned* to any department, we will be unable to put the data into the table because the w_dept field does not allow nulls.

3. Delete Anomaly

- If the corporation closes the department F890 at some point in the future, deleting the rows with w_dept as F890 will also erase the information of employee Mike, who is only assigned to this department.

➢ Hence, to overcome these anomalies we need to normalize the data.

Advantages of Normalization

- Minimize redundancy from table or relation.
- Manage insertion, modification and deletion of anomalies or inconsistencies.
- Helps to maintain data consistencies in the database.
- Puts the data into the form that is more able to accurately accommodate or sustain the change.
- Avoids unnecessary coding, etc.

1. First Normal Form(1NF)

- 1NF says each attribute of a table must have atomic values.
- If a relation contain composite or multi-valued attribute, it violates first normal form(1NF) or a relation is in first normal form if it does not contain any composite or multi-valued attribute.
- A relation is in first normal form if every attribute in that relation is **singled valued attribute**.

- Let us take an UNF(**Un – Normalized Form)** relation containing multi – valued attributes:

Student_Table

S.N.	Name	Address	Phone_No.
1	Shiva	Brt – 1	9812777809, 9842500021
2	Parbati	Brt – 2	9816949887
3	Radha	Brt – 3	9815599837
4	Krishna	Brt – 4	9816949847

- Now, the 1NF of the above UNF table is as follows:

Contd. ….

Student_Table

S.N.	Name	Address	Phone_No._1	Phone_No._2
1	Shiva	Brt – 1	9812777809	9842500021
2	Parbati	Brt – 2	9816949887	Null
3	Radha	Brt – 3	9815599837	Null
4	Krishna	Brt – 4	9816949847	Null

STUD_NO	STUD_NAME	STUD_PHONE	STUD_STATE	STUD_COUNTRY
1	RAM	9716271721, 9871717178	HARYANA	INDIA
2	RAM	9898297281	PUNJAB	INDIA
3	SURESH		PUNJAB	INDIA

Table 1

Conversion to first normal form

STUD_NO	STUD_NAME	STUD_PHONE	STUD_STATE	STUD_COUNTRY
1	RAM	9716271721	HARYANA	INDIA
1	RAM	9871717178	HARYANA	INDIA
2	RAM	9898297281	PUNJAB	INDIA
3	SURESH		PUNJAB	INDIA

Table 2

Second Normal Form (2NF)

- A relation is said to be in 2NF:
  - If and only if it is already in 1NF.
  - If a table has some attributes which are partially dependent on the primary key of that table, then it is not in 2NF.
  - Hence, Introduce Primary Key and foreign key to reduce functional dependencies by breaking a table to obtain 2NF table.
- Let us consider the following table.

Teacher	Subject	Age	Address
Ram	Maths	26	Pokhara
Ram	Physics	26	Pokhara
Sita	English	25	Palpa
Hari	Nepali	28	Butwal

1.Since, ***age and address*** depends upon the teacher we introduce a teacher's ID(T_id) as a primary key.

Table:1(Independent Table)

T_id	**Teacher**	**Age**	**Address**
1	Ram	26	Pokhara
2	Sita	25	Palpa
3	Hari	28	Butwal

Table : 2 (Dependent Table)

2. Since, ***Subject*** depends upon teacher,

T_id	**Subject**
1	Maths
1	Physics
2	English
3	Nepali

❖Problem

- Suppose if one teacher(Ram ,T_id =1) left from the organization ,then in Table 1 all the related information like Age and Address will be vanished/deleted which does effect on the Table 2 as well that means all (T_id =1) containing Subject Column will also be vanished this is called transitive Functional dependency.
- Hence, to solve this problem(Transitive Functional dependency) we shift towards 3NF resp.

Note:

1. In Table 1 (T_id)is act as a primary key where as in Table 2 it is act as a foreign key respectively.

➢ Third Normal Form (3NF)

- **A relation is said to be in 3NF if and only if:**

✓ **It should be in always in 2NF.**

✓ ***if a table contains transitive dependency, then it is not in 3NF and the table must be partitioned to bring it into 3NF. i.e remove functional dependency.***

For example: if a → b and b → c, then a → c.

i.e. if (a → b → c) form occurs in a table, then it is not in 3NF.

Contd. ….

Example: Consider the following relation which is not in 3NF:

Student_Table

S_ID	S_Name	Age	Gender	Hostel_Name
1	Shiva	21	M	Sunrise
2	Krishna	22	M	Sunrise
3	Radha	23	F	Suryajyoti
4	Parbati	24	F	Suryajyoti

- Here, we have:

  S_ID → S_Name

  S_ID → Age

  S_ID → Gender

  S_ID → Hostel_Name

  Gender → Hostel_Name

Contd. ….

- This last dependency Gender → Hostel_Name was not originally specified but to have derived it. This derived dependency is called a transitive dependency.
- In such a case, the relation should be broken into relations to make it in 3NF as follows:

S_ID	S_Name	Age	Gender
1	Shiva	21	M
2	Krishna	22	M
3	Mohan	23	M
4	Parbati	24	F

Gender	Hostel_Name
M	Sunrise
F	Suryajyoti

Conclusion

- A database anomaly usually occurs as a result of poor planning and the use of flat databases.
- In this article we learned about insert, delete and update anomalies as well as the circumstances that can lead to them.
- Anomalies are usually removed by normalizing the tables by splitting or joining them.
- Normalization provides structured data

❑ Transforming ER – Diagrams into Relations

- Transforming ER – Diagrams into normalized relations and then merging all the relations into one final set of relations can be accomplished within the following four steps:

I. Represent Entities

- Each entity type in the ERD becomes a relation. The identifier of the entity types becomes the primary key of the relation, and other attributes become non – prime key attributes of the relation.

II. Represent Relationships

- Each relationship in an ERD must be represented in the relational database design. How we represent a relationship depends on its nature. Relationships may be 1:1, 1:N, M:N, etc.

III. Normalize the Relations

- The relations created in steps 1 and 2 may have unnecessary redundancies. So, we need to normalize these relations to make them well – structured.

IV. Merge the Relations

- So far in database design, we have created various relations from both a bottom – up normalization of user views and from transforming one or more ERDs into set of relations. Across these different set of relations, there may be redundant relations that must be merged and re – normalized to remove the redundancies.

❑ Merging the Relations

- As a part of the logical database design, normalized relations likely have been created from a number of separated ERD and various user interfaces.
- Some of the relations may be redundant, so to merge the relations, we must define the objectives of removing the redundancies.

Example of merging the relations:

- Suppose that modelling a user interface or transforming an ERD results in the following 3NF relation:

  EMPLOYEE (<u>Emp_ID</u>, Name, Address, Phone_No.);

- Modelling a second user interface might result in the following relation:

  EMPLOYEE (<u>Emp_ID</u>, Name, Address, Job, Number_of_Year);

Contd. ….

- Because these two relations have the same primary key (Emp_ID) and describe the same entity, they should be merged into one relation.
- **The result of merging the relation is as follows:**

EMPLOYEE (<u>Emp_ID</u>, Name, Address, Phone_No., Job, Number_of_year);

- **Note that**, attribute that appears in both the relations (such as name in this example) appears only once in the merged relation.

Compiled by Bikash Magar || bikashtm17@gmail.com

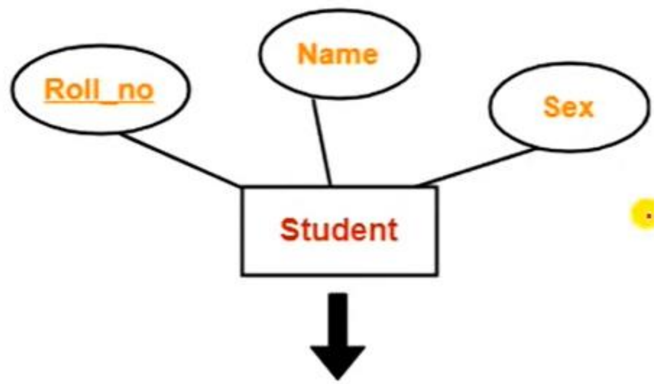
#Converting E-R Diagrams to Tables.

Rule1: Strong entity set with only simple Attributes.

- A strong entity set with only simple attributes will require **only one table** in relational model.

Note 1: Attributes of the table will be the attributes of the entity set.

Note 2: The primary key of the table will be the key attribute of the entity set.

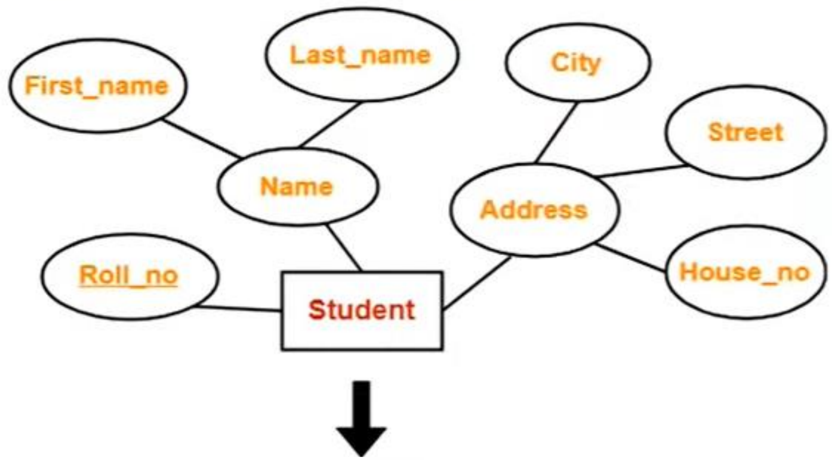


Roll_no	Name	Sex

Schema : Student (Roll_no , Name , Sex)

Rule2: Strong entity set with composite Attributes.

- A strong entity set with any number of composite attributes will require **only one table** in relational model.
- While transferring into table, simple attributes of the composite attribute are taken into account and not the composite attribute itself.



Roll_no	First_name	Last_name	House_no	Street	City

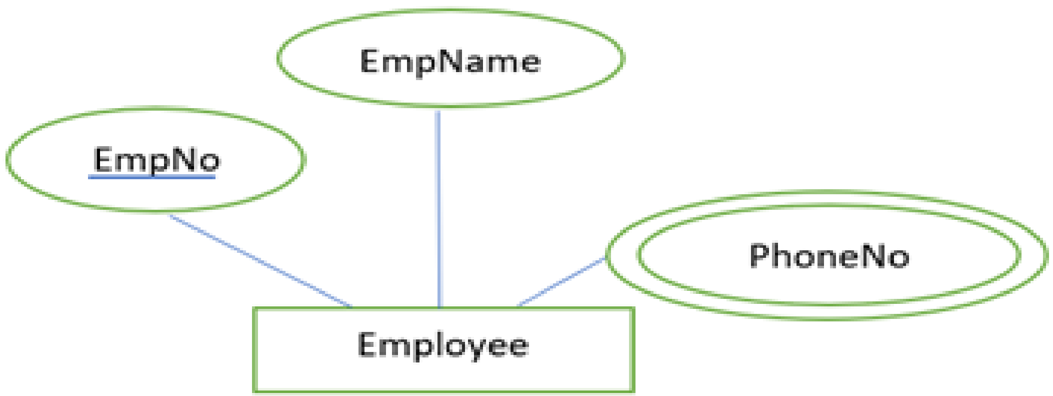
Schema : Student (Roll_no , First_na[illegible] , Last_name , House_no , Street , [illegible]

Rule3: Strong entity set with Multi valued Attributes.

- If an entity contains a multivalued attribute, we split the attributes into two relations in the relational model. One with key attribute and all simple attributes and other with key attribute and all multivalued attributes.
- For example, in the figure given below, *PhoneNo* is the multivalued attribute.

Note 1: One table will contain all the simple attributes with the primary key.

Note 2: Other table will contain the primary key and all the multi valued attribute.



- If we include the PhoneNo in the table with all other attributes, then for a single-valued tuple we may have multiple entries as shown in the table below.

Employee

EmpNo	EmpName	PhoneNo
1	A	9821
1	A	9780
2	B	1234
....		

Table: Duplicate values with multivalued attribute

However, to avoid duplicate values in the table, we split the attributes into two different relations as shown in the figure below

EmpNo	EmpName

EmpNo	PhoneNo

Rule 4:Translating Relationship set into a table.

- Attribute of the table are:

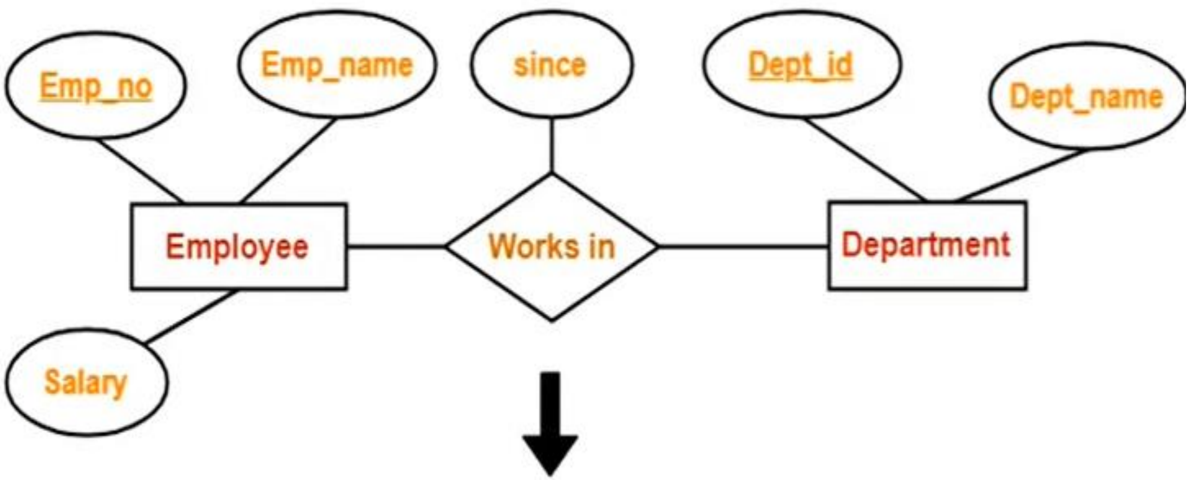
1. Primary key attributes of the participating entity sets.(The attributes are will become a foreign key here)
2. Its own descriptive attributes if any.

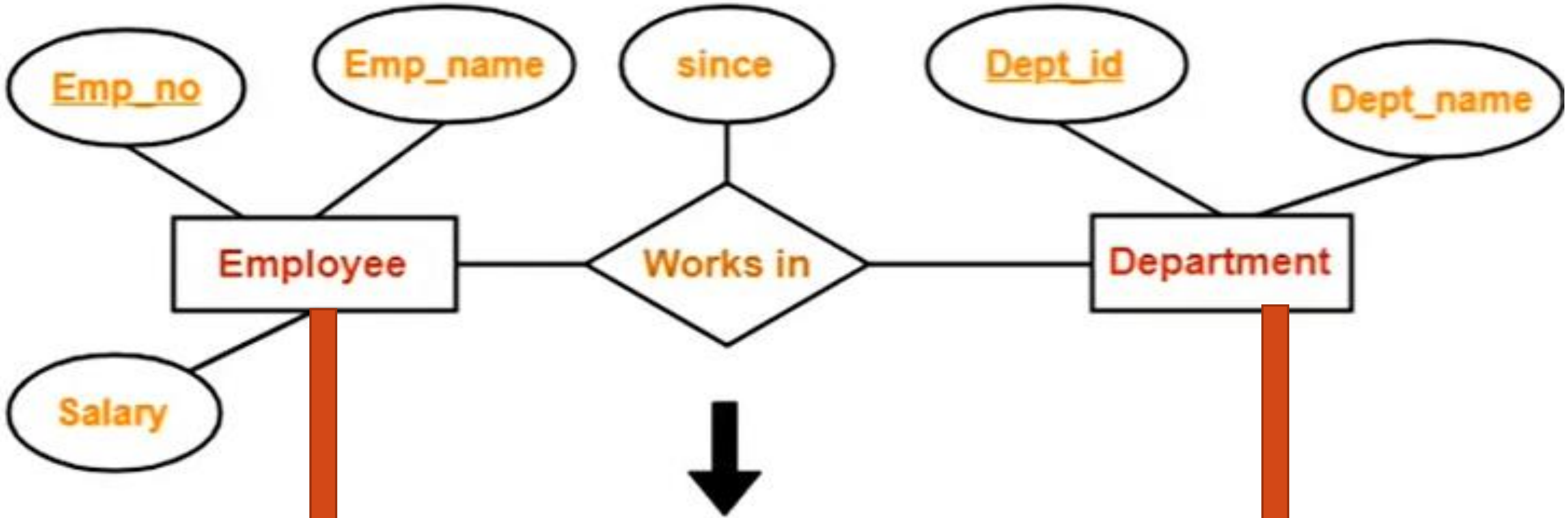
- For given ER-Diagram, three tables will be required in relational model.

  1.One table for entity set "Employee".

  2.One table for entity set "Department".

  3. One table for the Relationship set "Works in".





Emp_no	Emp_name	Salary

Emp_no (FK)	Dept_id (FK)	since

Dept_id	Dept_name

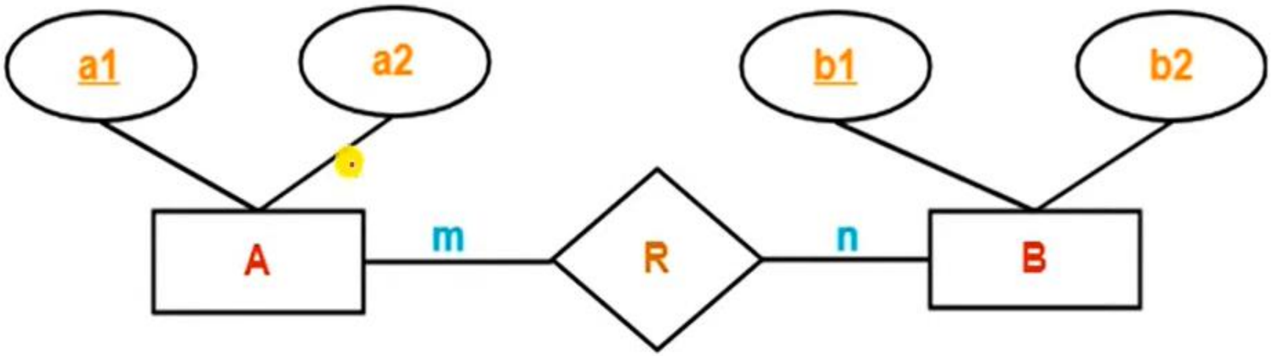
Schema : Works in (Emp_no , Dept_id , since)

Rule 5: For Binary relationships with cardinality Ratio.

- Case 1: Binary Relationship with cardinality ratio M:N.
- Case 2: Binary Relationship with cardinality ratio 1:N.
- Case 3: Binary Relationship with cardinality ratio M:1.
- Case 3: Binary Relationship with cardinality ratio 1:1.

Case 1: Binary Relationship with cardinality ratio M:N.

Case-1: For Binary Relationship with Cardinality Ratio m:n

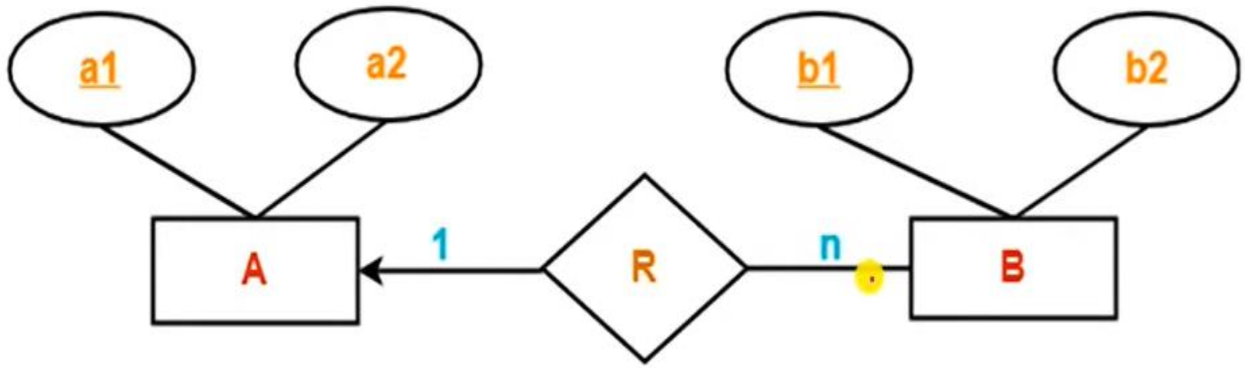


In ***Many-to-Many relationship***, **three tables** will be required-

1. A (a1 , a2)
2. R (a1 , b1)
3. B (b1 , b2)

Case 2: Binary Relationship with cardinality ratio 1:N.

Case-2: For Binary Relationship with Cardinality Ratio 1:n



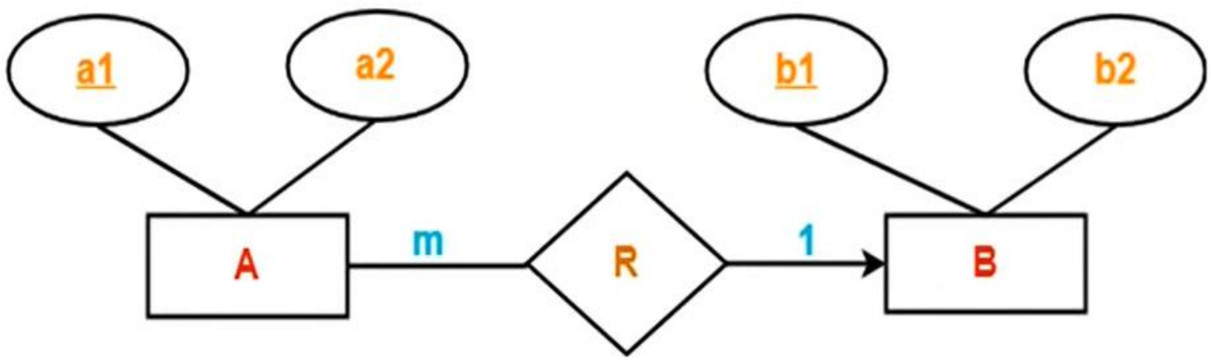
In ***One-to-Many relationship*, two tables** will be required-

1. A (<u>a1</u> , a2) FK
2. BR (<u>b1</u> , b2, a1)

<u>NOTE</u> - Here, combined table will be drawn for the entity set B and relationship set R.

Case 3: Binary Relationship with cardinality ratio M:1.

Case-3: For Binary Relationship with Cardinality Ratio m:1



In ***Many-to-One relationship***, **two tables** will be required-

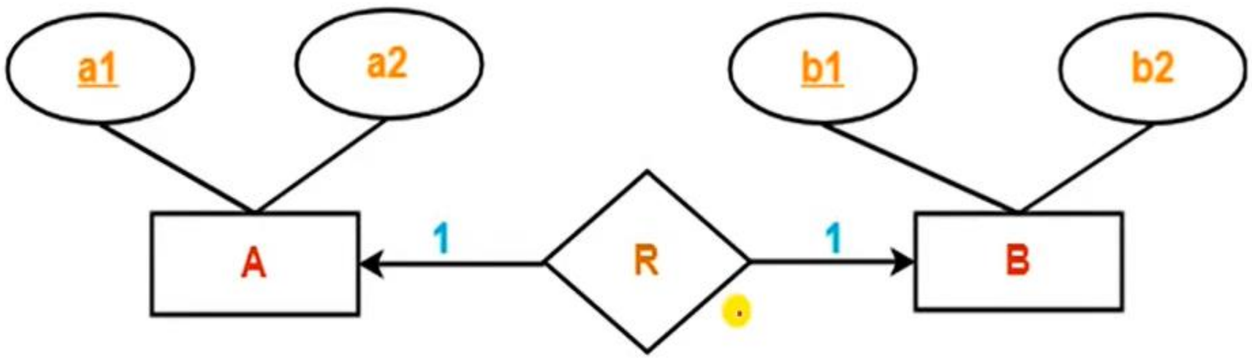
FK

1. AR (a1 , a2 , b1)
2. B (b1 , b2)

NOTE - Here, combined table will be drawn for the entity set A and relationship set R.

Case 4: Binary Relationship with cardinality ratio 1:1.

Case-4: For Binary Relationship with Cardinality Ratio 1:1



In ***One-to-One relationship***, **two tables** will be required. Either combine 'R' with 'A' or 'B'

Way-01:

FK

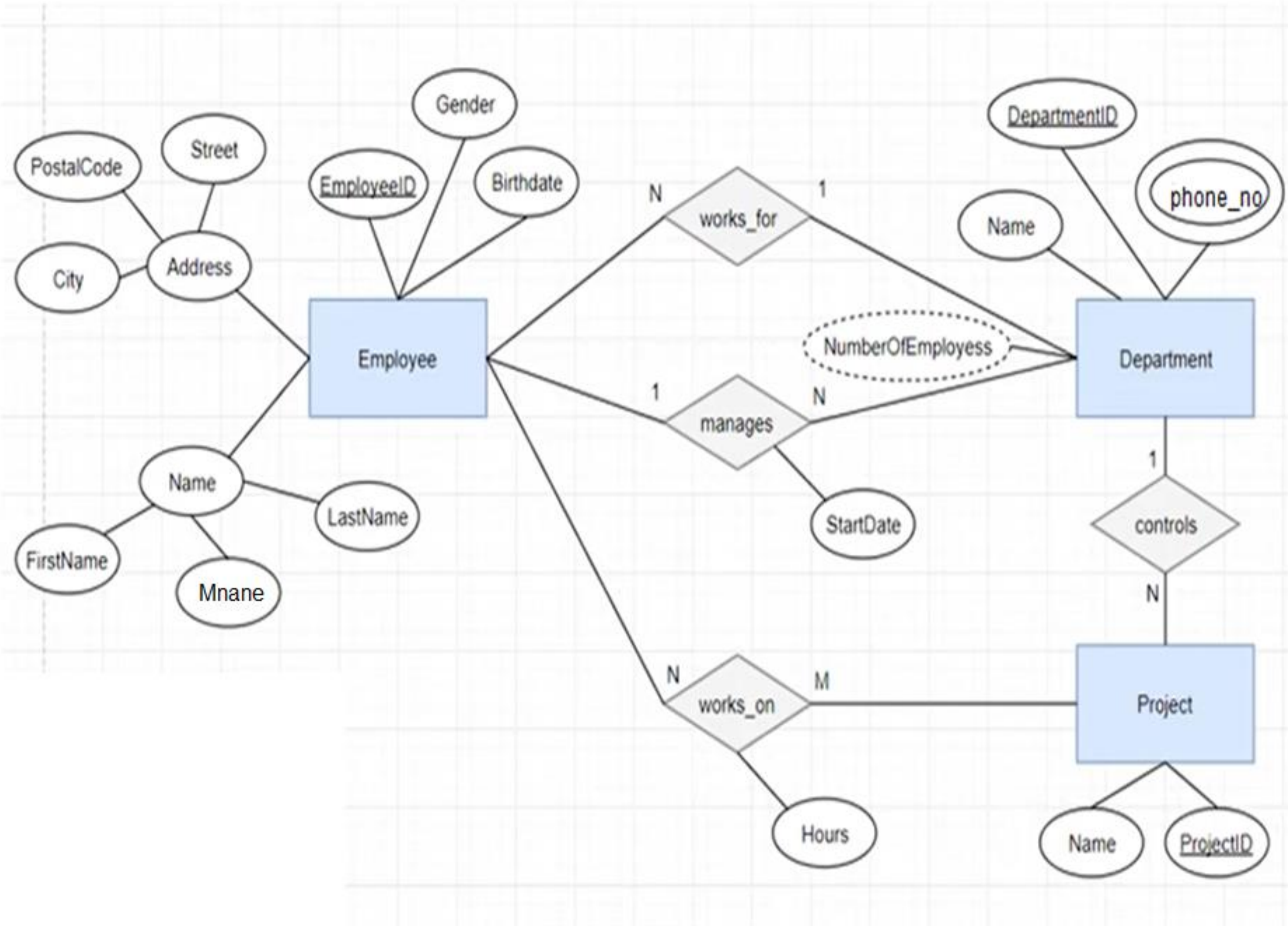
1. AR (a1 , a2 , b1)
2. B (b1 , b2)

Way-02:

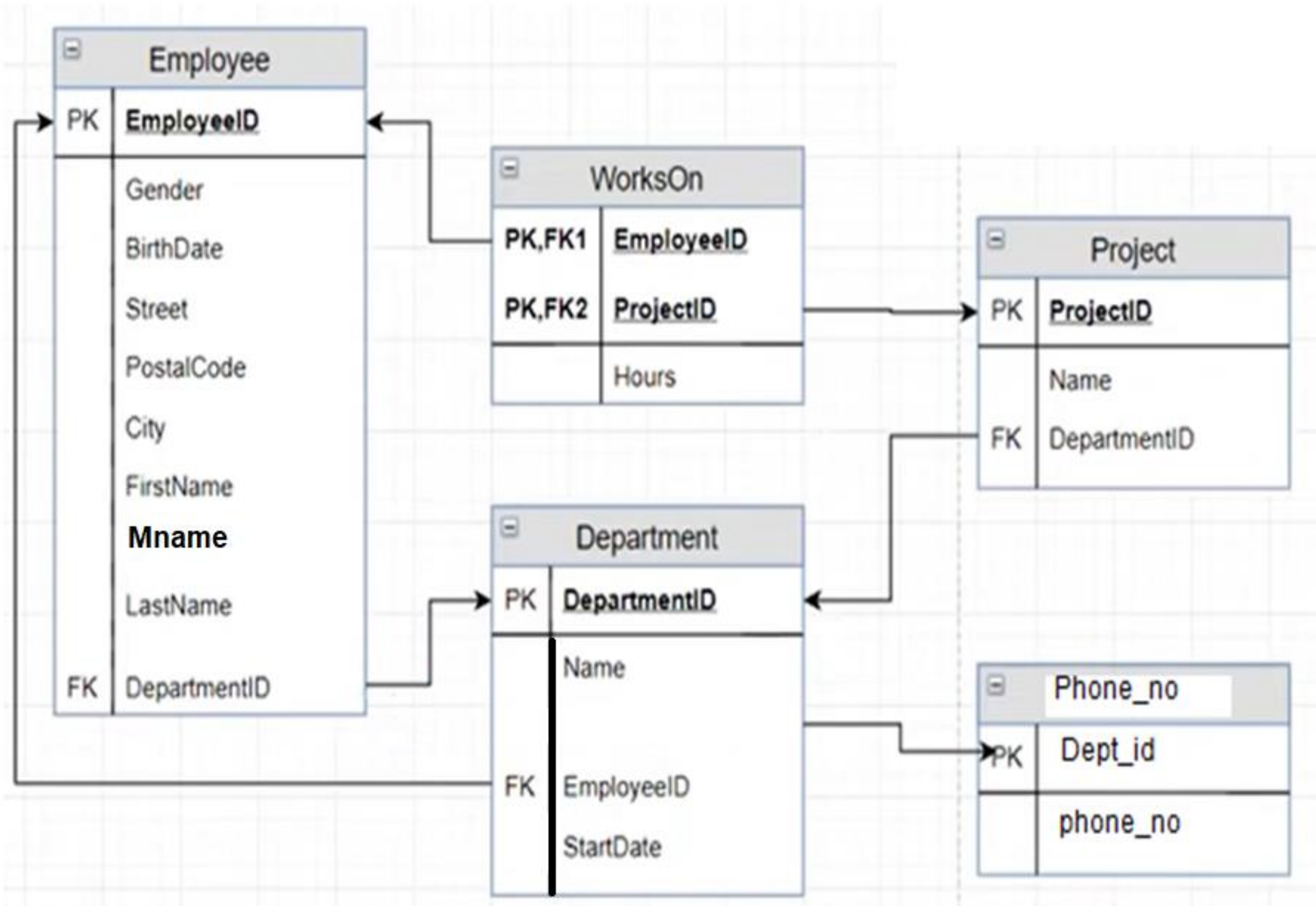
1. A (a1 , a2)

FK

2. BR (b1 , b2, a1)



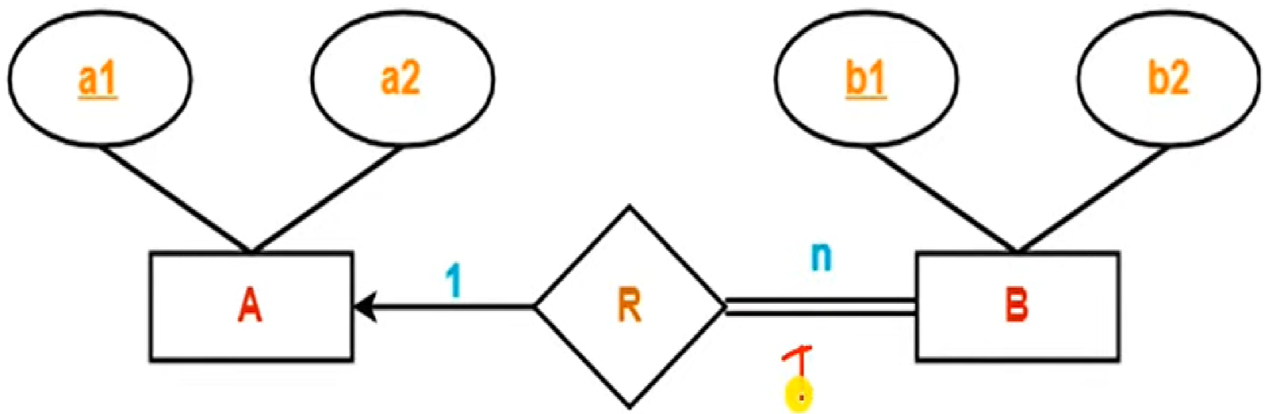
PostalCode
Street
Gender
EmployeeID
Birthdate
City
Address
Employee
Name
LastName
FirstName
Mnane
N
works_for
1
1
manages
N
StartDate
NumberOfEmployess
N
works_on
M
Hours
DepartmentID
phone_no
Name
Department
1
controls
N
Project
Name
ProjectID


Employee
PK EmployeeID
Gender
BirthDate
Street
PostalCode
City
FirstName
Mname
LastName
FK DepartmentID
WorksOn
PK,FK1 EmployeeID
PK,FK2 ProjectID
Hours
Project
PK ProjectID
Name
FK DepartmentID
Department
PK DepartmentID
Name
FK EmployeeID
StartDate
Phone_no
PK Dept_id
phone_no

Rule-6: For Binary Relationship with Both Cardinality Constraints and Participation Constraints

- Cardinality constraints will be implemented as discussed in Rule-5.
- Because of the *total participation constraint*, foreign key acquires **NOT NULL** constraint i.e. now foreign key can not be null.
- Two Cases:
  - **Case-1:** For Binary Relationship with Cardinality Constraint and Total Participation Constraint from One Side
  - **Case-2:** For Binary Relationship with Cardinality Constraint and Total Participation Constraint from Both Sides

Case-1: For Binary Relationship with Cardinality Constraint and Total Participation Constraint from One Side



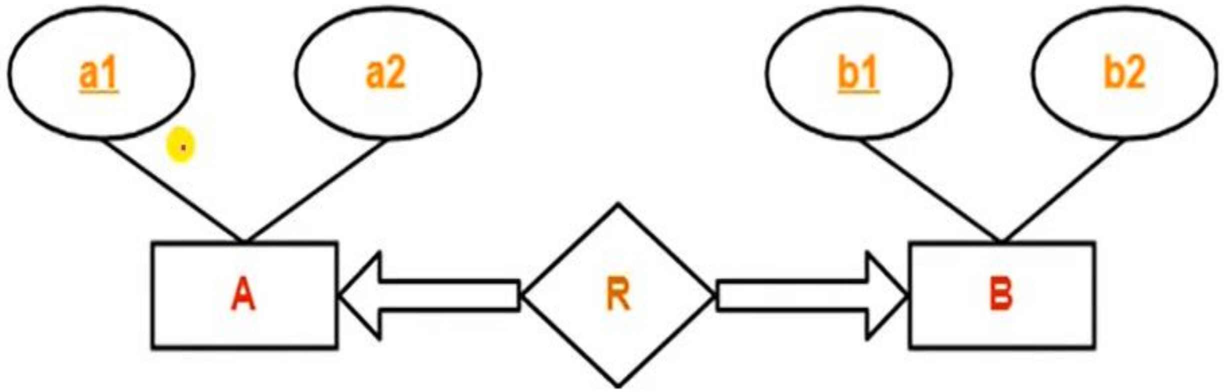
- Because cardinality ratio = 1 : n , we will combine the entity set B and relationship set R.
- Then, **two tables** will be required-

1. A (<u>a1</u> , a2)
2. BR (<u>b1</u> , b2 , a1) FK

Because of total participation, ***foreign key a1*** has acquired ***NOT NULL constraint***, so it can't be null now.

Case-2: For Binary Relationship with Cardinality Constraint and Total Participation Constraint from Both Sides

- If there is a key constraint from both the sides of an entity set with total participation, then that binary relationship is represented using **only single table.**

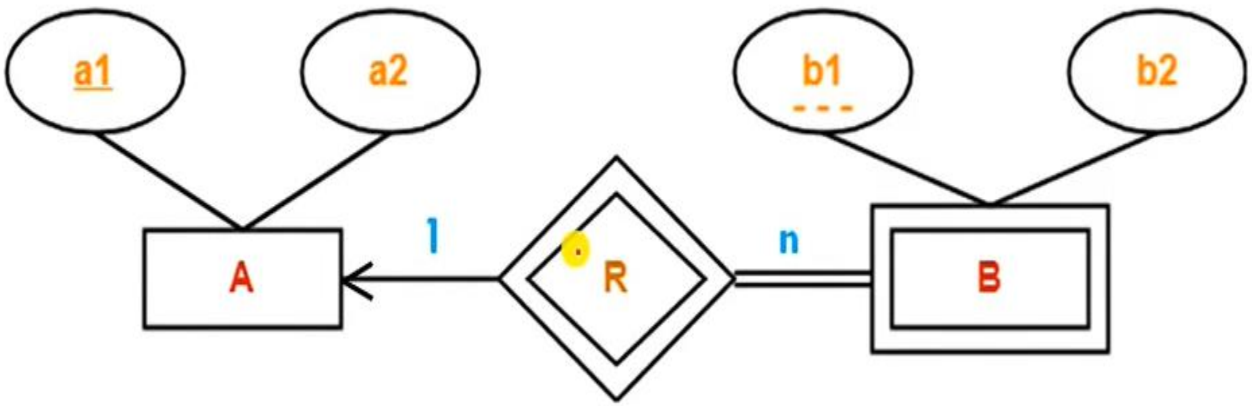


- Here, Only **one table** is required.

1. **ARB (<u>a1</u> , a2 , <u>b1</u> , b2)**

Rule-7: For Binary Relationship with Weak Entity Set

- Weak entity set always appears in association with identifying relationship with **total** participation constraint and there is always **1: n** relationship from identifying entity set to weak entity set.



- Here, **two tables** will be required-
  1. **A (<u>a1</u> , a2)**
  2. **BR (<u>a1</u> , <u>b1</u> , b2)**

Physical File and Database Design

- Designing physical(Physical files **contain the actual data that is stored on the system, and a description of how data is to be presented to or received from a program**) files and databases requires certain information that should have been collected and produced during prior SDLC phases.
- ***This information includes the following:***

✓ Normalized relations, including volume estimates(Space to store and maintain our database)

✓ Definitions of each attribute.

✓ Descriptions of where and when data are used: entered, retrieved, deleted, and updated (including frequencies).

✓ Expectations or requirements for response time and data integrity.

✓ Descriptions of the technologies used for implementing the files and database so that the range of required decisions and choices for each is known.

- Thus, we begin the physical design phase by addressing the **design of physical fields** and the **design of physical tables** for each attribute in a logical data model.

Designing Fields

- A field is the smallest unit of application data recognized by system software, such as programming language or DBMS.
- An attribute from a logical database model may be represented by the several fields.

For example: A student name attribute in a normalized student relation might be represented as three fields: last name, first name, and middle name.

- The basic decisions we must take in specifying each field concern the type of data (or storage type) used to represent the filed and data integrity controls for the field.
- **Calculated Field:** A field that can be derived from other database fields is called a calculated field (**or a computed field or a derived field**). **For example**, an invoice may include a total due field, which represents the sum of the amount due on each item on the invoice.

Field

Table

Record

Value

id	ISSN-L	ISSNs	PublisherId	Journal_Title
0	2056-9890	2056-9890	1	Acta Crystallographica Section E Crystallographic Communications
1	2077-0472	2077-0472	2	Agriculture
2	2073-4395	2073-4395	2	Agronomy
3	2076-2615	2076-2615	2	Animals
4	2076-3417	2076-3417	2	Applied Sciences
5	2306-5354	2306-5354	2	Bioengineering
6	2079-7737	2079-7737	2	[illegible]
7	2079-6374	2079-6374	2	[illegible]

Designing Physical Tables

- A relational database is a set of related tables, related by the foreign keys referencing primary keys.
- In logical database design, we group into a relation those attributes that concern some unifying, normalized business concept, such as customer, product, or employee.
- In contrast, a physical table is a named set of rows and columns that specifies the fields in each row of the table.
- A physical table may or may not correspond to one relation, whereas normalized relations possess properties of well – structured relations.
- **The design of a physical table has two goals, different from those of normalization: efficient use of secondary storage, and data processing speed.**

Compiled by Bikash Magar || bikashtm17@gmail.com

❖ Designing forms and reports in system analysis and design:

- Forms and reports serve as the user interface to interact with a system and present information in a structured and meaningful way.
- Here are some guidelines for designing forms and reports in the context of system analysis and design:

❑ Forms Design:

1. User-Centered Design:

- Understand the needs and preferences of the end users.
- Involve end users in the design process to ensure the form meets their requirements.

2. Clarity and Simplicity:

- Keep forms simple and easy to understand.
- Use clear and concise labels for fields.
- Arrange elements logically to guide the user through the form.

3. Consistency:

- Maintain consistency in terms of layout, fonts, and colors.
- Use standard conventions for form elements (e.g., use radio buttons for exclusive choices).

4. Input Validation:

- Include validation checks for data accuracy.
- Provide meaningful error messages to guide users when mistakes occur.

5. Navigation:

- Design an intuitive flow within the form.
- Include navigation elements like tabs or buttons for easy movement between sections.

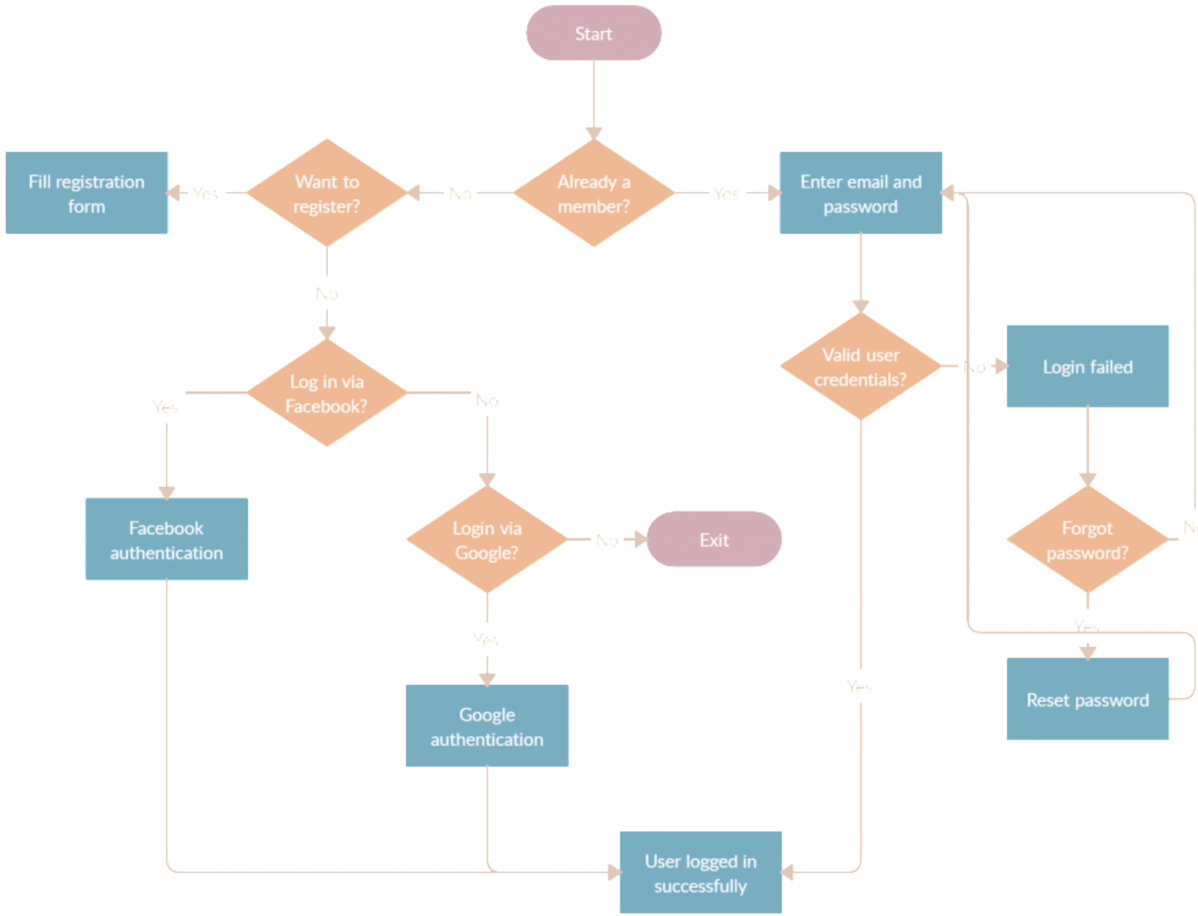
6. Responsive Design:

- Ensure the form is usable on different devices and screen sizes.

7. Feedback:

- Provide feedback to users after form submission.
- Use visual cues to indicate mandatory fields or successful submissions.

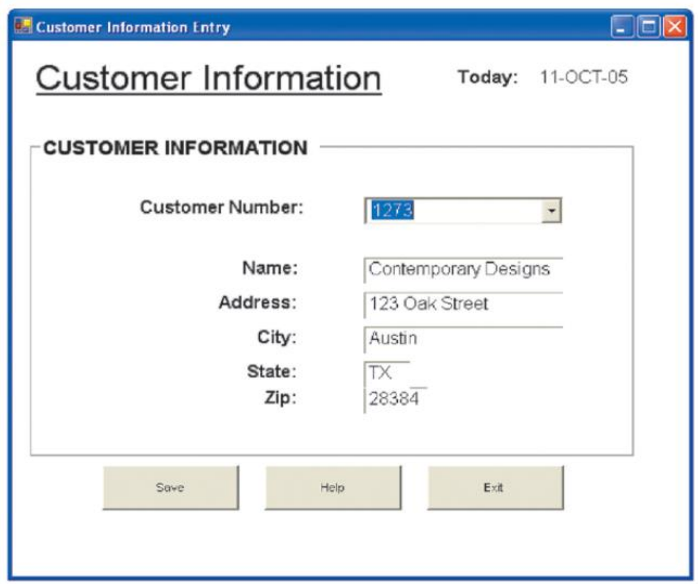
Intututive flow



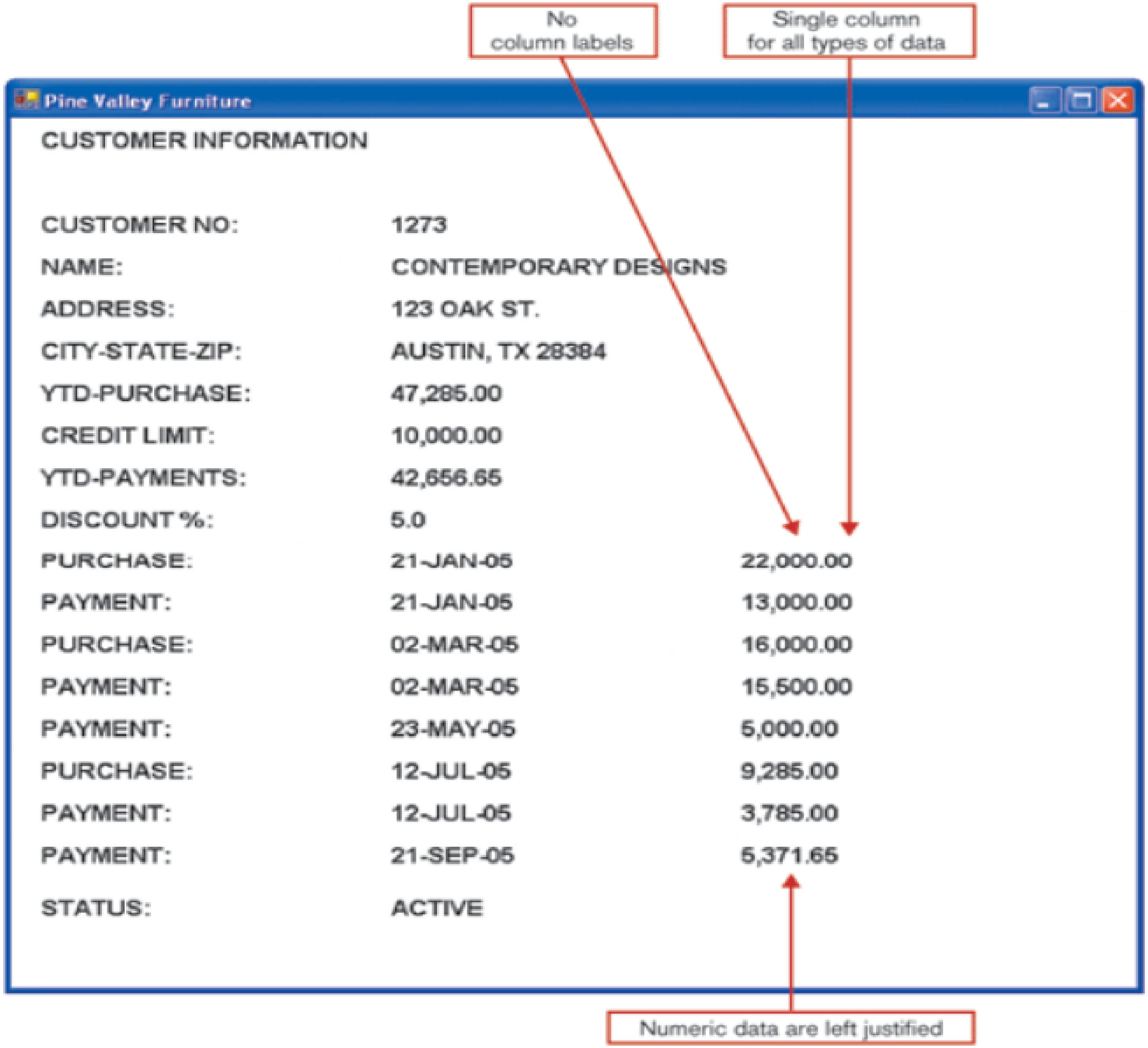
Design Forms & Reports

Form

- A business document that contains some pre – defined data, and may include some areas where additional data are to be filled in is called a form.
- In a database context, a form is a window or screen that contains numerous fields ,or spaces to enter data. Each field holds a field label so that any user who views the form gets an idea of its content.
- A form is more user friendly than generating queries to create tables and insert data into fields. An instance of a form is typically based on one database record.



Visual Basic and other development tools provide computer aided GUI form and report generation.

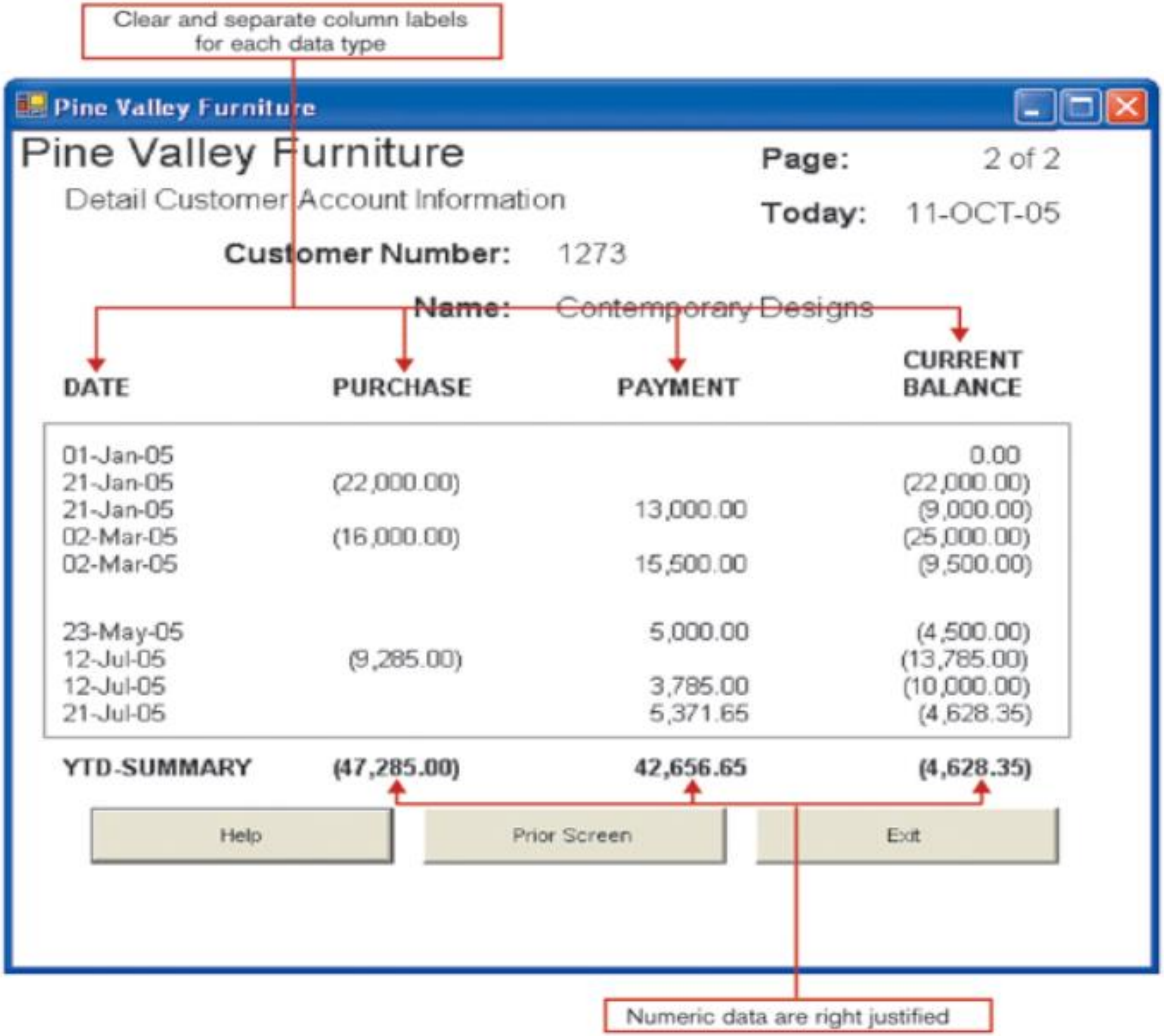


Pine Valley Furniture

CUSTOMER INFORMATION

CUSTOMER NO:	1273	
NAME:	CONTEMPORARY DESIGNS	
ADDRESS:	123 OAK ST.	
CITY-STATE-ZIP:	AUSTIN, TX 28384	
YTD-PURCHASE:	47,285.00	
CREDIT LIMIT:	10,000.00	
YTD-PAYMENTS:	42,656.65	
DISCOUNT %:	5.0	
PURCHASE:	21-JAN-05	22,000.00
PAYMENT:	21-JAN-05	13,000.00
PURCHASE:	02-MAR-05	16,000.00
PAYMENT:	02-MAR-05	15,500.00
PAYMENT:	23-MAY-05	5,000.00
PURCHASE:	12-JUL-05	9,285.00
PAYMENT:	12-JUL-05	3,785.00
PAYMENT:	21-SEP-05	5,371.65
STATUS:	ACTIVE	

A poor table design



A better table design

Report

- A business document that contains only pre – defined data, and is a passive document used solely for reading or viewing purposes is called a report.
- A report typically contains data from many unrelated records or the transactions.

Common Types of Reports

- Scheduled: produced at predefined time intervals for routine information needs
- Key-indicator: provide summary of critical information on regular basis
- Exception: highlights data outside of normal operating ranges
- Drill-down: provide details behind summary of key-indicator or exception reports
- Ad-hoc: respond to unplanned requests for non-routine information needs

Types of Business Report

- ✓ **Scheduled Report**

- Reports produced at pre – defined intervals – daily, weekly, or monthly to support the routine information needs of an organization are called scheduled reports.

- ✓ **Key – Indication Report**

- Reports that provide a summary of critical information in a recurring basis are called key – indication reports.

- ✓ **Exception Report**

- Reports that highlight data which are out of the normal operating range are called exception reports.

- ✓ **Drill – Down Report**

- Reports that provide details behind the summary values on a key indicator or exception reports are called drill – down reports.

- ✓ **Ad – hoc Report**

- These types of reports contain unplanned information requests in which information is gathered to support a non – routine decision.

The Process of Designing Forms and Reports

1. **User-focused activity**

The process of designing forms and reports is a user – focused activity that typically follows a prototyping approach.

2. Follows a prototyping approach

- ✓ It starts with collecting the initial requirements, and then structuring and refining those information into an initial prototype. Structuring and refining requirements are completed independently of the users.
- ✓ After reviewing the prototype or evaluating the prototype, users may accept the design or request for the changes to be made.
- ✓ If the changes are needed, we will repeat the construction – evaluation – refinement cycle until the design is accepted. The cycle might go through the multiple iterations.

3. Requirements determination:

Here, process of designing forms and reports arises the requirements determination, such as :

- ✓ Who will use the form or report?
- ✓ What is the purpose of the form and report?
- ✓ When is the form and report needed or used?
- ✓ When does the form or report need to he delivered and used?
- ✓ How may people need to use or view the form or report?, etc.

- ***The deliverables and outcomes from the process of designing forms and reports are the design specifications, and they are as follows:***
- **Narrative Overview**: It contains a general overview of the characteristics of the target users, tasks, system, and environmental factors in which the form or report will be used.
- **Sample design**: image of the form (from coding sheet or form building development tool) In this section, a sample design of the form is drawn. The design may be hand – drawn using a coding sheet, although in most instances, it is developed using CASE or standard development tools.
- **Assessment**: measuring test/usability results (consistency, sufficiency, accuracy, etc.) It provides all the testing and usability assessment information. Procedures for assessing designs are described in this section.

Formatting Forms and Reports

- *The process or guidelines of formatting forms and reports are:*

✓ **Meaningful Title**

- Clear and specific titles describing the content and use of forms and reports.
- Revision data, current data, valid date, etc.

✓ **Meaningful Information**

- Only needed information should be displayed and it should be provided in usable format without modifications.

✓ **Balance the layout**

- Information should be balanced on the screen or page, and sufficient spacing and margins should be used.

✓ **Design an easy navigation system**

- Clearly show how to move forward and backward.
- Clearly show where we are, and notify user when on the last page of a multiple sequence.

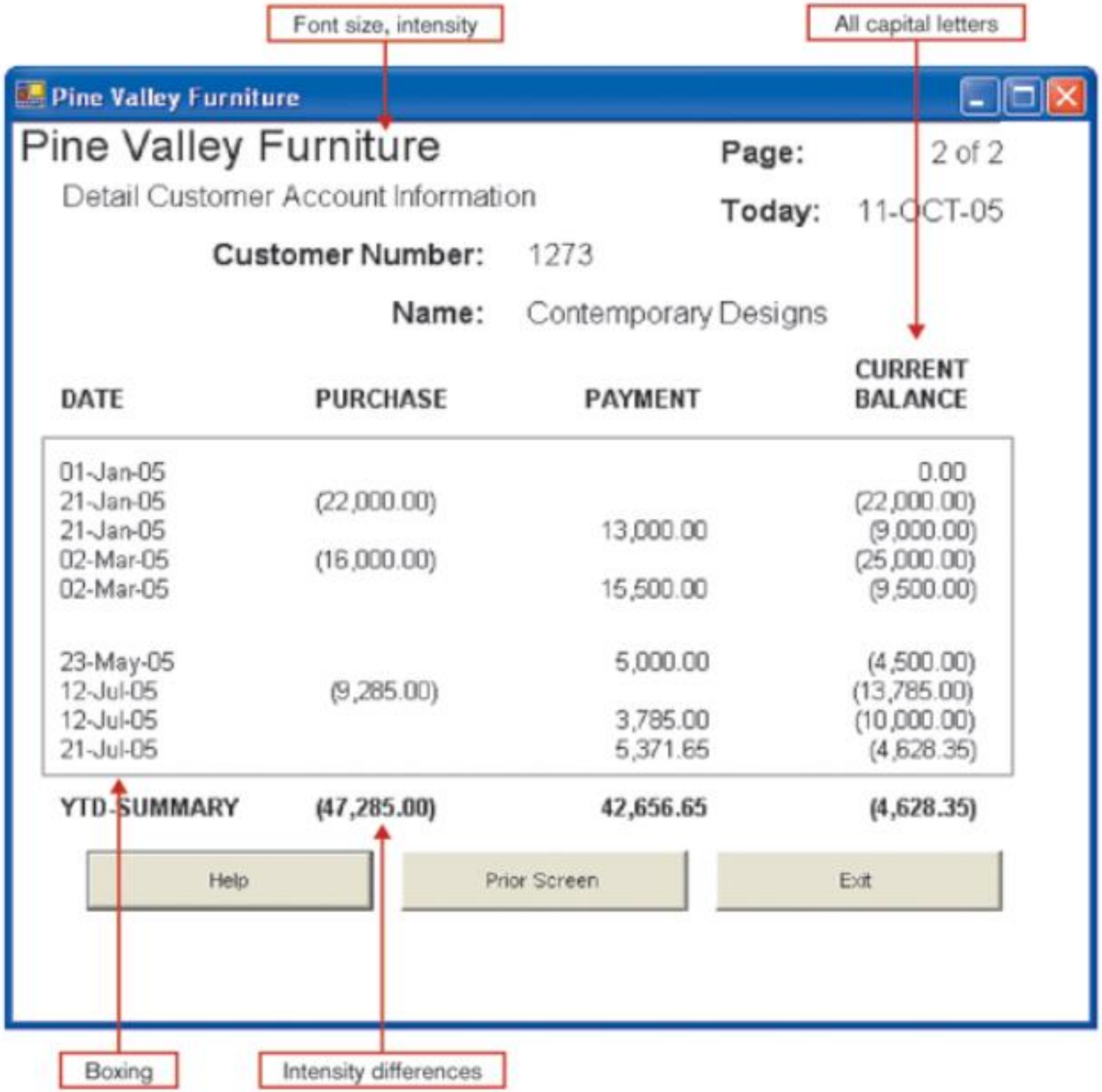
Contd. ….

✓ **Highlighting Information**

- Notifying users of errors in data entry or processing.
- Providing warnings to users regarding possible problems such as unusual data values or an unavailable device.
- Methods of highlighting includes: blinking and audible tones, colour differences, size differences, font differences, underlining, all capital letters, etc.
- Draw attention to keywords, commands, high priority messages, unusual data values.

Methods for Highlighting

- Blinking
- Audible tones
- Intensity
- Reverse video
- Boxing
- Underlining
- Intensity differences
- Size differences
- Font differences
- Underlining
- All capital letters
- Offset positions of nonstandard information

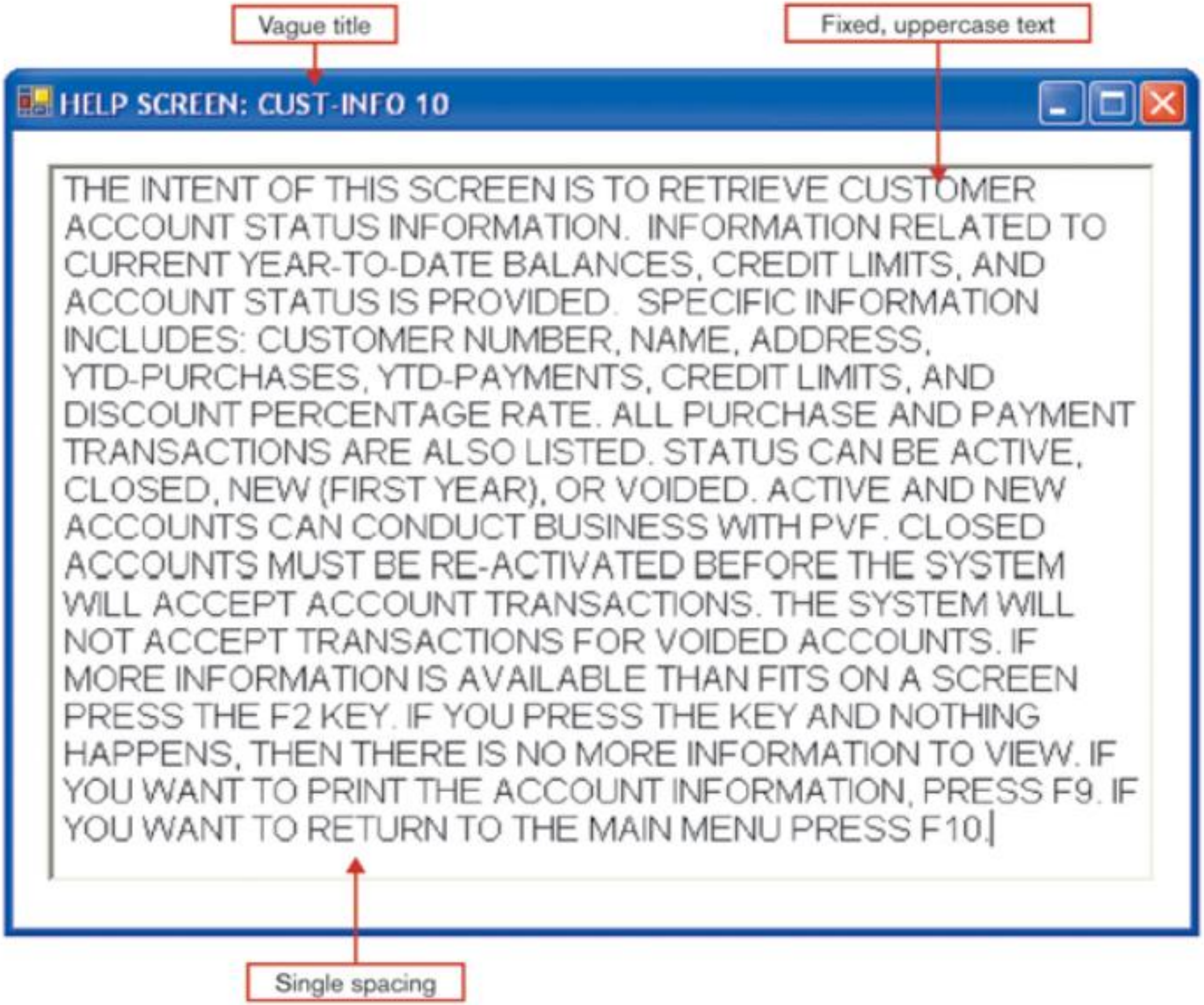


Highlighting can include use of upper case, font size differences, bold, italics, underline, boxing, and other approaches.

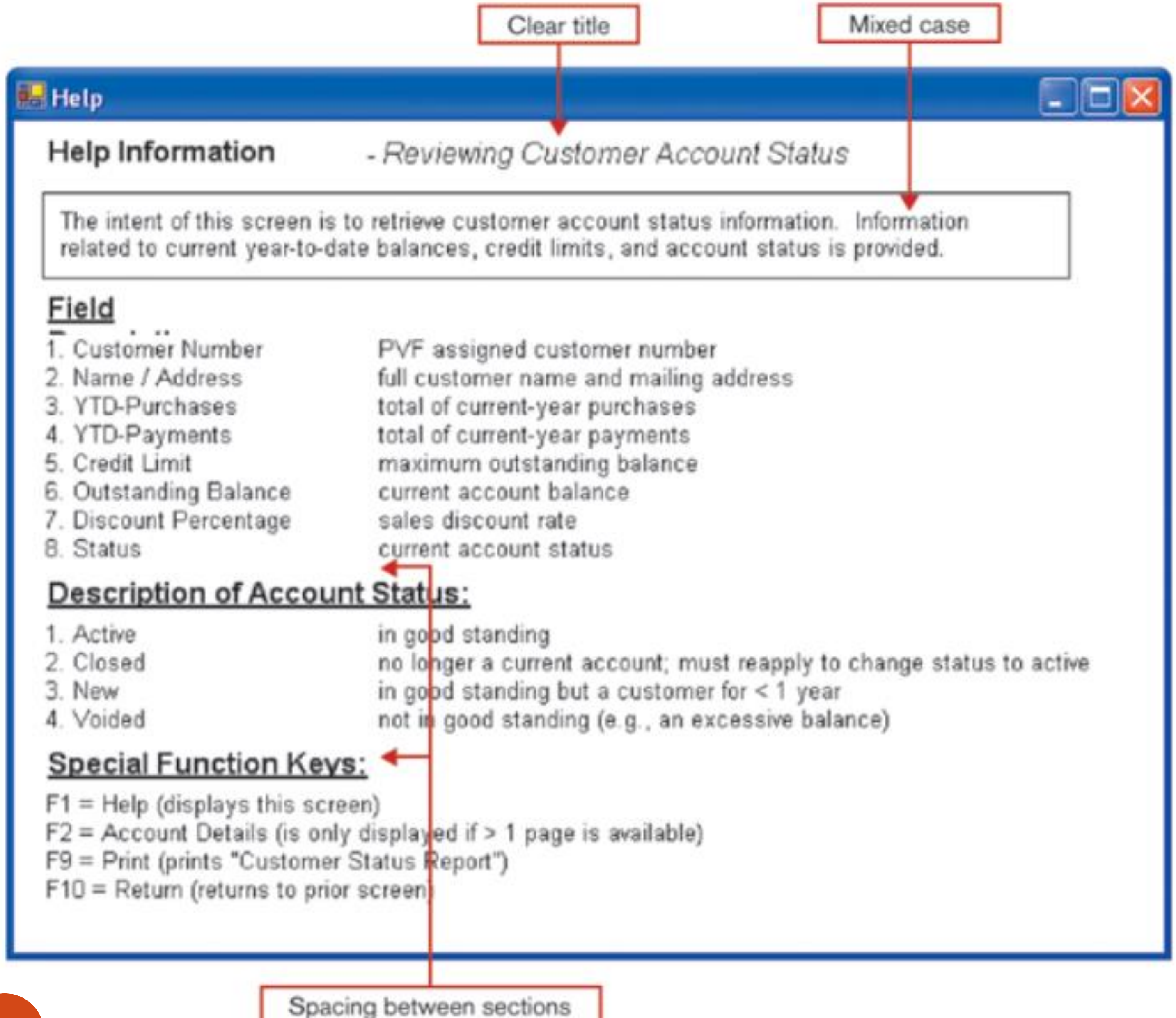
✓ **Displaying Text**

❑Guidelines for Displaying Text

- In business related system, textual output is becoming increasingly important as text – based applications such as email, bulletin, and information services are more widely used, etc.
- Case: mixed upper and lower case, use conventional punctuation
- Spacing: double spacing if possible, otherwise blank lines between paragraphs.
- Justification: left justify text, ragged right margins.
- Hyphenation: no hyphenated words between lines
- Abbreviations: only when widely understood and significantly shorter than full text



A poor help screen design



Help

Help Information *- Reviewing Customer Account Status*

The intent of this screen is to retrieve customer account status information. Information related to current year-to-date balances, credit limits, and account status is provided.

Field

1. Customer Number	PVF assigned customer number
2. Name / Address	full customer name and mailing address
3. YTD-Purchases	total of current-year purchases
4. YTD-Payments	total of current-year payments
5. Credit Limit	maximum outstanding balance
6. Outstanding Balance	current account balance
7. Discount Percentage	sales discount rate
8. Status	current account status

Description of Account Status:

1. Active	in good standing
2. Closed	no longer a current account; must reapply to change status to active
3. New	in good standing but a customer for < 1 year
4. Voided	not in good standing (e.g., an excessive balance)

Special Function Keys:

F1 = Help (displays this screen)
F2 = Account Details (is only displayed if > 1 page is available)
F9 = Print (prints "Customer Status Report")
F10 = Return (returns to prior screen)

A better help screen design

Contd. ….

- **Color vs. No Color**
- Color is a powerful tool for the designer in influencing the usability of an information system.
- When applied appropriately, color provides many potential benefits to forms and reports, such as draws attention to warnings, strikes the eyes, emphasizes the logical organization of information.
- Problems from using color are such as color – blindness, resolution may degrade with different displays, printing or conversion to other media may not easily translate the information, etc.

Color vs. No Color

- Benefits from Using Color
- Soothes or strikes the eye
- Accents an interesting display
- Facilitates subtle discriminations in complex
- Emphasizes the logical organization of information
- Draws attention to warnings
- Evokes more emotional reactions

Problems from Using Color

- Color pairings may wash out or cause problems for some users
- Resolution may degrade with different displays
- Color fidelity may degrade on different displays
- Printing or conversion to other media may not easily translate

Assessing Usability

- Overall evaluation of how a system performs in supporting a particular user for a particular task
- There are three characteristics
- There are three characteristics

- Speed — Can you complete a task efficiently?
- Accuracy— Does the output provide what you expect?
- Satisfaction -Do you like using the output?

Guidelines for Maximizing Usability

- **Consistency**: of terminology, formatting, titles, navigation, response time
- **Efficiency**: minimize required user actions
- **Ease**: self-explanatory outputs and labels
- **Format**: appropriate display of data and symbols
- **Flexibility**: maximize user options for data input according to preference

❑ Assessing Usability

- There are many factors to consider when we design forms and reports. The objective for designing forms and reports, and human – computer interactions (HCI) is assessing usability.
- *Usability typically refers to the following three characteristics:*

✓ **Speed:** Can we complete a task efficiently?

✓ **Accuracy:** Does the system provide what we expect?

✓ **Satisfaction:** Do we like using the system?

- In other words, usability means that our designs should assist, not hinder, user performance.
- Thus, usability refers to an overall evaluation of how a system performs in supporting a particular user for a particular task.

Contd. ….

General Design Guidelines for Usability of Forms and Reports

Usability Factor	Guidelines for Achievement of Usability
Consistency	Consistent use of terminology, abbreviations, formatting, titles, and navigation within and across outputs. Consistent response time each time a function is performed.
Efficiency	Formatting should be designed with an understanding of the task being performed and the intended user. Text and data should be aligned and sorted for efficient navigation and entry. Entry of data should be avoided where possible (e.g., computing rather than entering totals).
Ease	Outputs should be self-explanatory and not require users to remember information from prior outputs in order to complete tasks. Labels should be extensively used, and all scales and units of measure should be clearly indicated.
Format	Information format should be consistent between entry and display. Format should distinguish each piece of data and highlight, not bury, important data. Special symbols, such as decimal places, dollar signs, and ±signs, should be used as appropriate.
Flexibility	Information should be viewed and retrieved in a manner most convenient to the user. For example, users should be given options for the sequence in which to enter or view data and for use of shortcut keystrokes, and the system should remember where the user stopped during the last use of the system.

Designing Interfaces and Dialogues

- The process of designing interfaces and dialogues is similar to the process of designing forms and reports.
- The process of designing interfaces and dialogues is a user – focused activity. This means that we follow the prototyping methodologies of iteratively collected information, constructing a prototype, assessing usability, and making refinements.
- To design usable interfaces and dialogues, we must answer the same who, what, when, where, and how questions used to guide the design of forms and reports.
- Thus, this process of designing interfaces and dialogues is parallel to that of designing forms and reports.
- *The deliverables and outcomes of designing interfaces and dialogues follow the following three sections:*

✓ *Narrative overview, Sample design, and Testing and usability assessment.*

Compiled by Bikash Magar ||
bikashtm17@gmail.com

Interaction Methods and Devices

- An interface is a method by which users interact with the computer information system.
- The **human – computer interface (HCI)** defines the ways in which users interact with an information system.

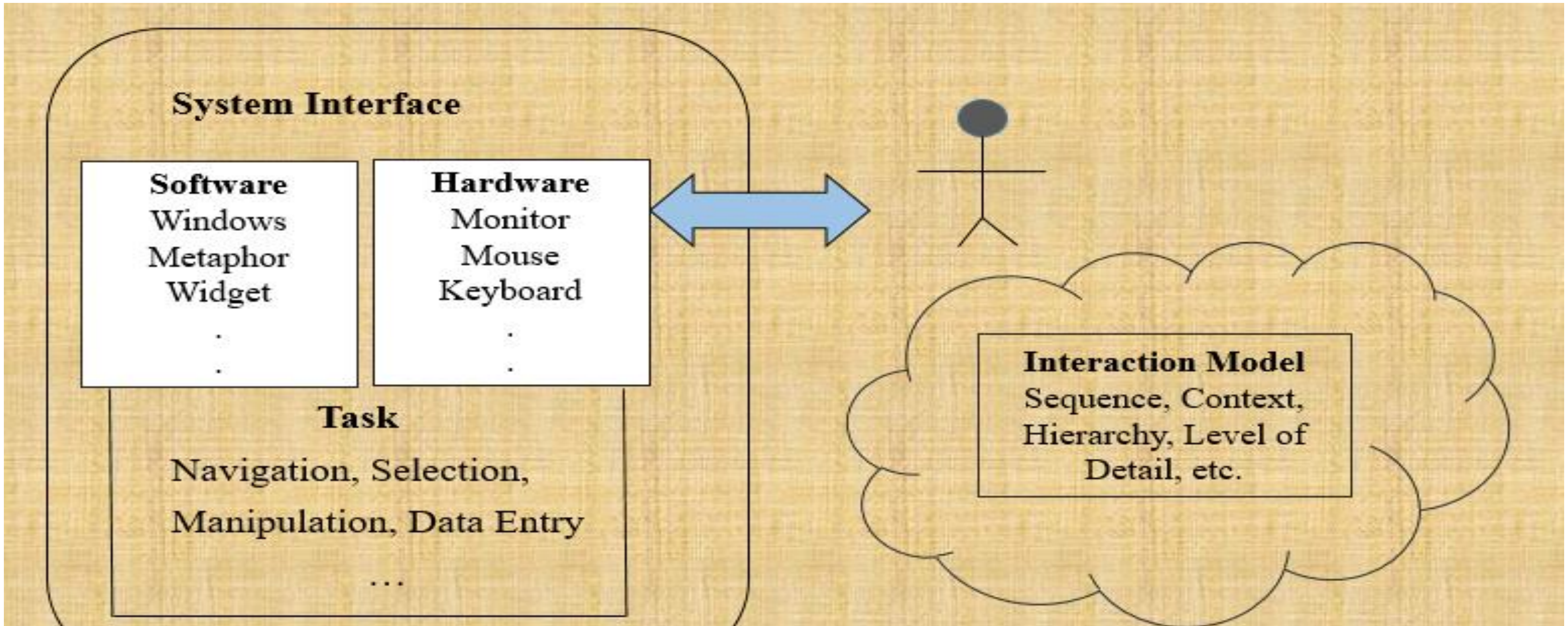


Fig: Human Computer Interaction

Compiled by Bikash Magar || bikashtm17@gmail.com

Contd. ….

- *There are various methods of interaction and some major of them are discussed below:*

Menu Interaction

- In this method, a user selects a command from the list of possible menu. It is often the case that another screen object is selected at the same time and the command operates on that objects. For example, to delete a file, user selects the file then selects the delete command.

Command Language Interaction

- In this method, the user issues a special command and associated parameters to instruct the system what to do. For example, to delete a file, the user issues a delete command with the filename as a parameter.

Form Interaction

- Here a user fills in the fields of a form. Some fields may have associated menus and the form may have action buttons that when pressed cause some action to be initiated. This is simple data entry and easy to learn but it takes up a lot of space in the screen.

Contd. ….

➢ Object – based Interaction

- Here, symbols are used to represent commands or functions. The most common method for implementing object – based interaction is making the use of icons. Icons are the graphical symbols that represent the specific function within a system.

➢ Natural Language Interaction

- In this method, the user issues a command in natural language. It is accessible to causal users and requires more typing. Natural language understanding systems are unreliable. For example, WWW information retrieval systems.

➢ Direct Manipulation Interaction

- In this method, the user interacts directly with objects on the screen. It is fast and intuitive interaction, and easy to learn. But, it may be hard to implement, only suitable where there is a visual metaphor for the tasks and the objects.

Designing Interfaces and Dialogues in Graphical Environments

- Graphical User Interface (GUI) environments have become the standard method for human – computer interaction (HCI).
- Although all of the interface and the dialogue design guidelines presented previously apply to designing GUI, additional issues that are unique to these environments must be considered.
- *Here, some of them are discussed below:*

Graphical interface design issues

- When designing HUI for an operating environment such as MS Windows, numerous factors must be considered.
- Some factors are common to all GUI environments where others are specific to single environment.
- ***Here, the effective GUI designer would play the role to design an interface and dialogue effectively as follows:***

✓ The first step should be an obvious one which says that become an expert user of the GUI environment.

✓ The second step focuses on understanding the available resources and how they can be used.

Contd. ….

Dialogue design issues

- When designing a dialogue, our goal is to establish the sequence of displays (full screen) that users will encounter when working with the system.
- The dialogue designing process may face the issues like what, where, when, how, etc.
- Within many GUI environments, encountering process can be a bit more challenging due to the GUI's ability to suspend the activities and switch the another application or task.
- Thus, these types of issues must be resolved to obtain good dialogue designing in the graphical environments.

Compiled by Bikash Magar ||
bikashtm17@gmail.com