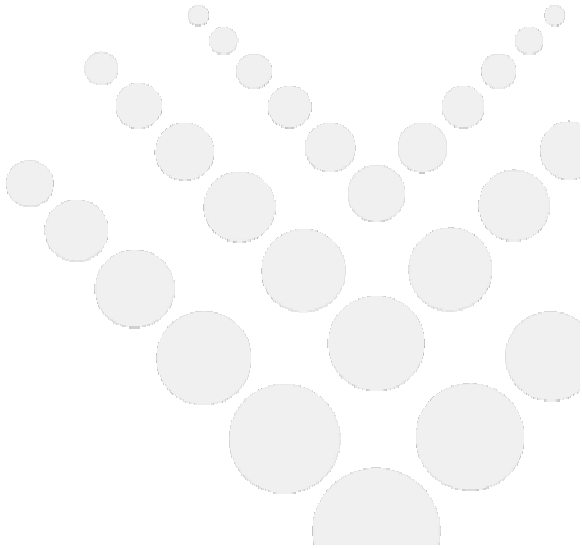
VINSYS
IT SERVICES INDIA PVT. LTD.

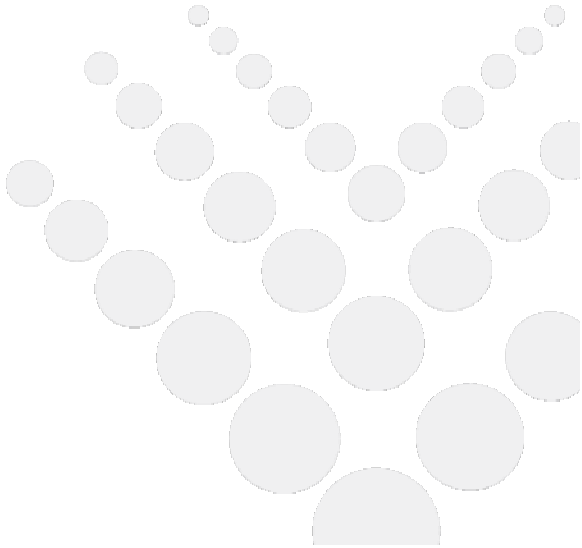
DBMS



Agenda

VINSYS
IT SERVICES INDIA PVT. LTD.

- Data and Storage.
- Drawbacks with File Systems.
- Introduction to DBMS and its benefits.
- Multi-tier Architecture
- Data Models
  - Hierarchical Model.
  - Network Model.
  - Relational Data Model.
- Relational Data Model concepts.
- Codd's 12 rules.



Data and its storage

VINSYS
IT SERVICES INDIA PVT. LTD.

- We require reliable data to work with and hence it becomes essential to store the data.
- We often have to store the data as it is crucial in business decisions.
- There are 2 ways to store data in computer:
  - Traditional Approach (File Systems).
  - Database Management Systems (DBMS).
- A File System is described by a very simple data storage model
  - In this, all the information is stored in a plain text file, one record per line.
  - Each record is divided into fields using delimiters or are delimited using fixed column positions.

Drawbacks with File Systems

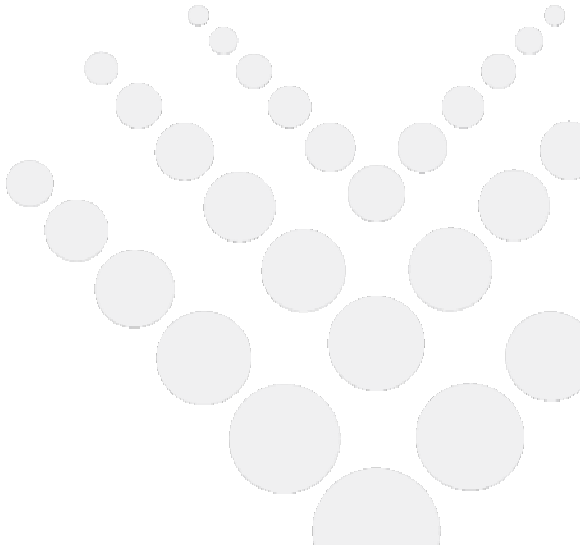
VINSYS
IT SERVICES INDIA PVT. LTD.

Drawbacks	Description
Data Dependency	Data dependency means that the application program is dependant on the data. So any modifications made on the data affects the application program using it, forcing it to be rewritten.
Redundancy	Redundancy is the repetition of data. Data duplication was allowed in the traditional approach. Data redundancy caused data inconsistency to occur in the data.
Sharing	Data Sharing was difficult or impossible with traditional systems.
Security	Data stored on the computer must be secured. It was difficult to enforce independent information security in traditional approach.
Transaction Control	Transaction is unit of work. If one is performing some work (for e.g. inserting some records) other persons should not be able to use that particular data. There was no way to isolate independent units of work in traditional approach.

Introduction to DBMS

VINSYS
IT SERVICES INDIA PVT. LTD.

- DBMS is **Data Base Management System.**
- It is a program that lets one or more computer users create and access data in a database.
- It allows data definition, storage, and management of data in the centralized pool that can be shared by many users.

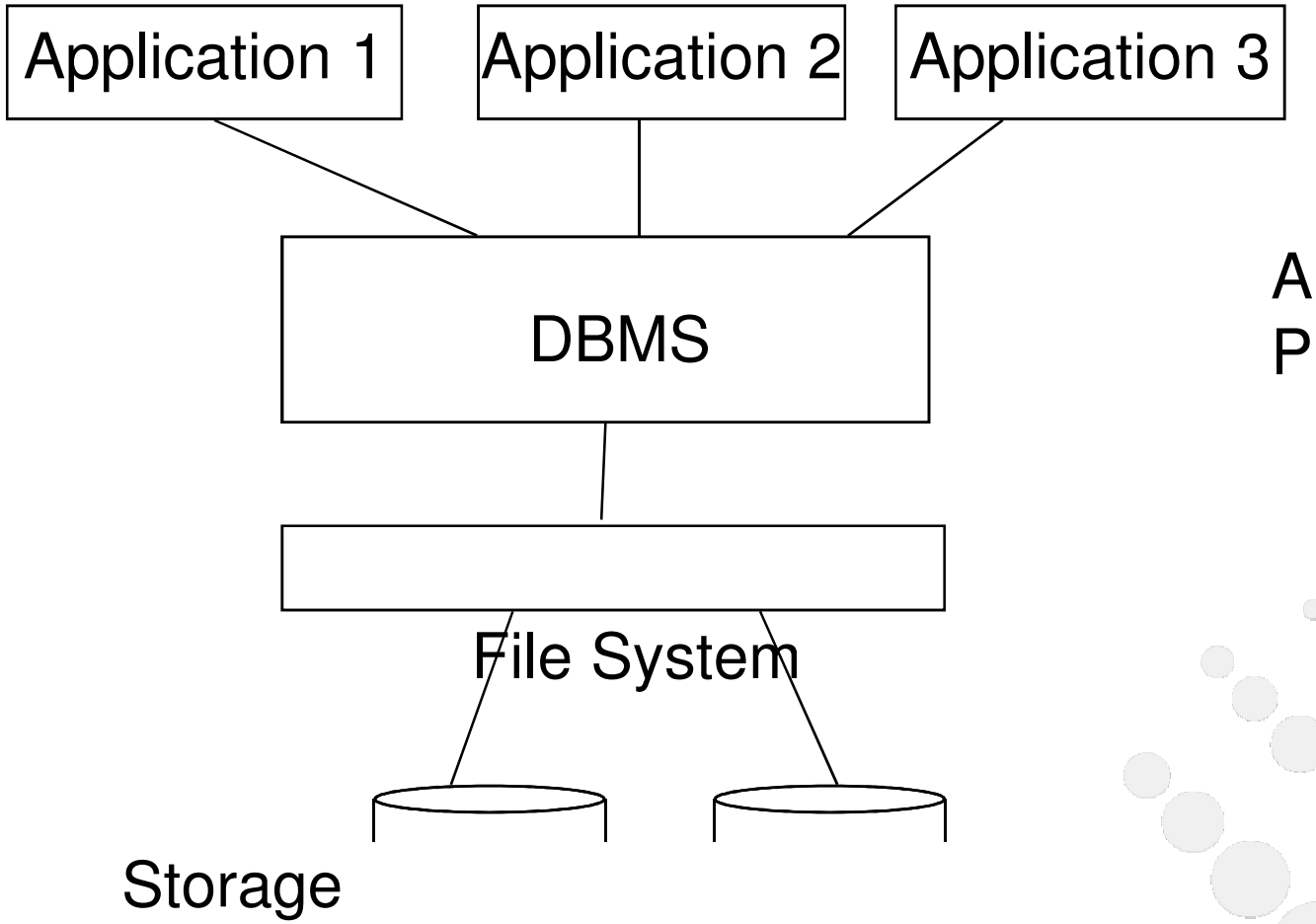


Benefits of DBMS

VINSYS
IT SERVICES INDIA PVT. LTD.

- Controls Data Concurrency
  - Databases allow multiple users to access the same database simultaneously.
- Avoids Data Redundancies
  - A properly set-up database minimizes data redundancy. It will avoid data repetition in the databases.
- Ensures Data Validity
  - Databases allow you to set up rules that ensure that data remains consistent when data is added or modified.
- Ensures Data Sharing
  - Since it allows multiple users to access the data, sharing data becomes easier in database.
- Ensures Data Security
  - Data is secured as different privileges and roles can be assigned to each user.

Application Programs using DBMS Services



Application Programs

Multi-tier Architecture

- Multi-tier architecture is also known as *n-tier architecture.*
- It is a client-server architecture in which an application is executed by more than one distinct software agent.
  - For example: An application that uses middleware to service data requests between a user and a database employs multi-tier architecture.
- The most widespread use of "multi-tier architecture" refers to three-tier architecture.

Three-tier Architecture

Presentation Tier

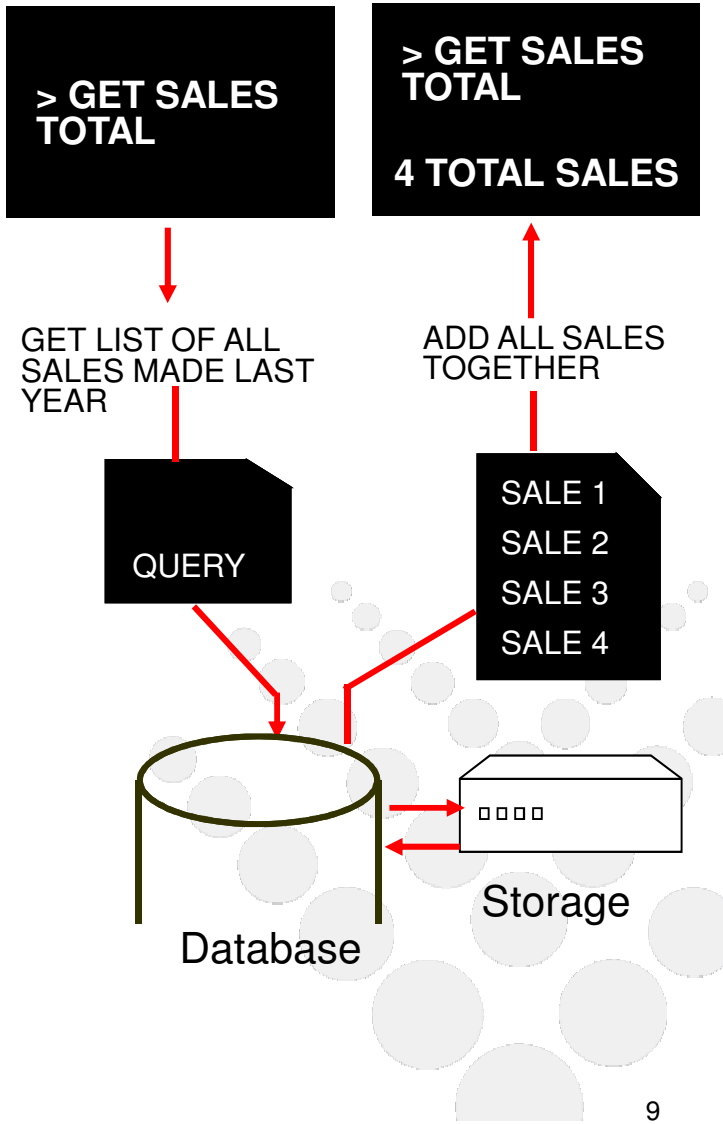
The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.

Logic Tier

This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.

Data Tier

Here Information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.

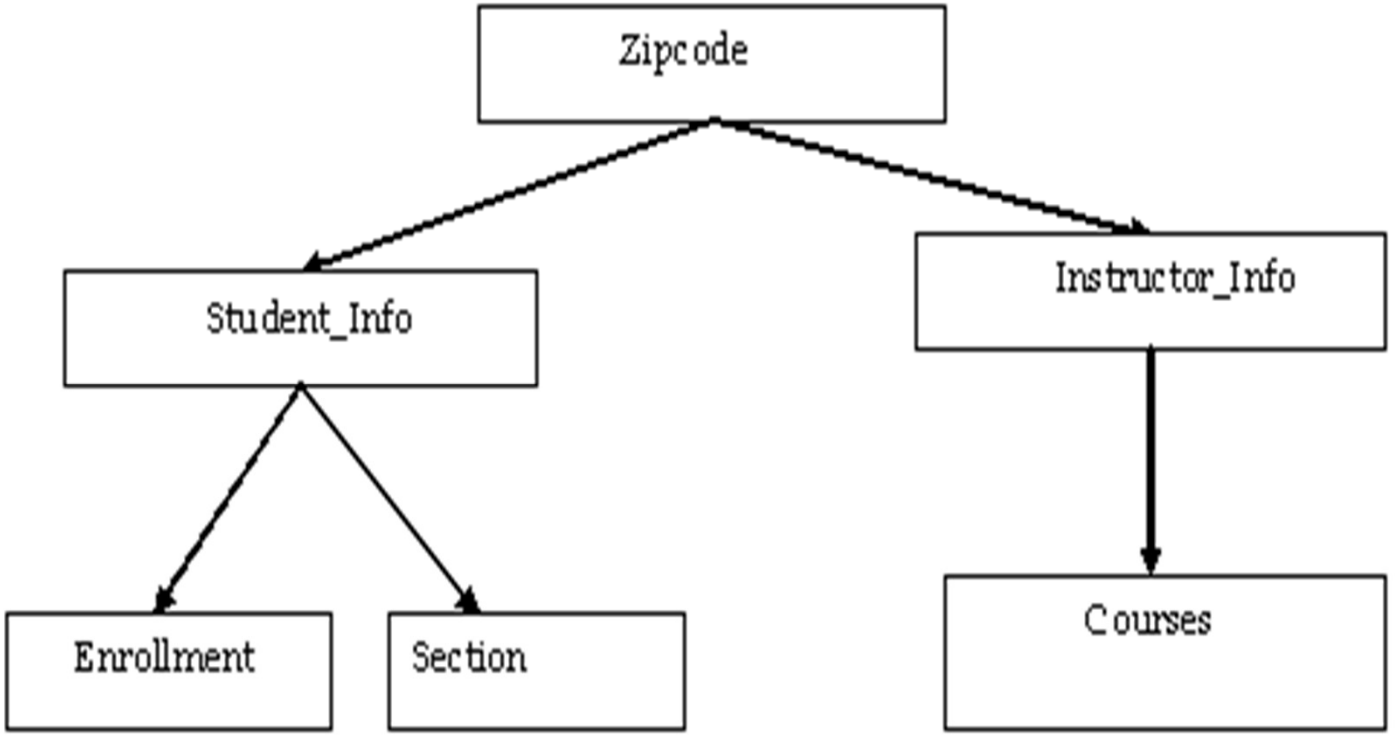


Data Models

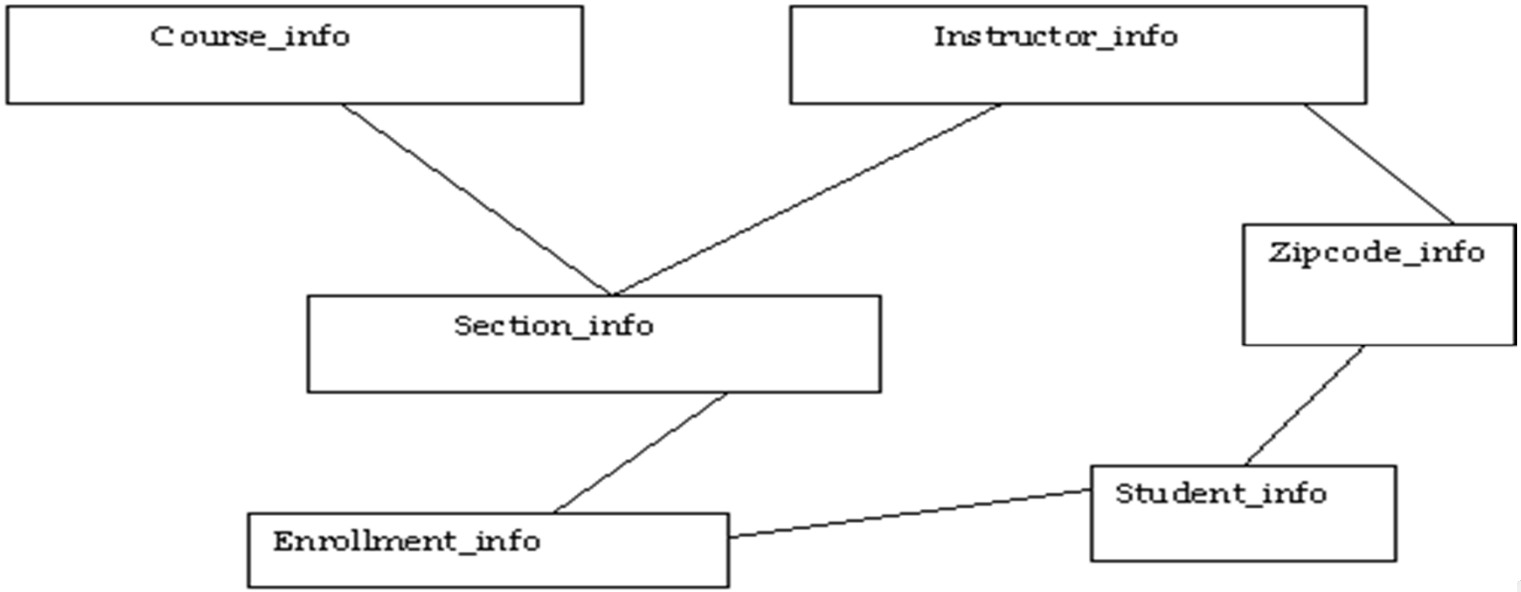
- **Data model** is a model that describes in an abstract way how data is represented in a business organization, an information system or a database management system.
- Following are the different data models available

Type of Data Model	Description
Hierarchical Model	Well suited for data which are related hierarchically.
Network Model	Database model conceived as flexible way of representing objects and their relationships.
Relational Model	Data is represented in the form of two-dimensional tables.

Hierarchical Data Model



Network Data Model



Relational Data Model

VINSYS
IT SERVICES INDIA PVT. LTD.

- It is based on Relational algebra i. e. mathematical theory of relations.
- It was first described by E.F. Codd in 1970 and then modified by IBM.
- It presents data in form of tables.
- Rules of Relational Data Model
  - It defines tables which contains Rows and Columns.
  - It presents a set of rules to ensure the integrity of the database.
  - In Relational model a table is also a relation.
  - Row is known as a tuple.
  - Column is known as an attribute.
- Dr. Codd modified his model by defining 12 rules (Codd's Rules), that a DBMS must meet in order to be considered a relational database.

Basic Terms and Definitions

VINSYS
IT SERVICES INDIA PVT. LTD.

- The term '**database**' has many interpretations - one definition is that database is a 'collection of persistent data‘.
- A **relational database** is one in which the data consists of a '**collection of tables related to each other through common values**‘.
- A **relational database management system (RDBMS)** uses matching values in multiple tables to relate the information in one table with the information in the other table.
- The presentation of data as tables is a logical construct - it is independent of the way the data is physically stored on disk.

Basic Terms and Definitions (Contd …)

VINSYS
IT SERVICES INDIA PVT. LTD.

- The two most prominent characteristics of a relational database are
  - Data stored in tables.
  - Relationships between tables.
- A **table** (entity or relation) is a collection of rows and columns.
- A **row** (record or tuple) represents a collection of information about a separate item (e.g., a customer).
- A **column** (field or attribute) represents the characteristics of an item (e.g., the customer's name or phone number).
- A **relationship** (join) is a *logical link* between two tables.

Codd's 12 rules

VINSYS
IT SERVICES INDIA PVT. LTD.

Rule No.	Rule	Description
# 1	Information rule	• Database must consist of tables related to each other and data should be stored in the form of tables only.
# 2	Guaranteed access Rule	• The data can be accessed by specifying the table name and the columns that define the primary key. Primary key ensures that each value is unique and accessible.
# 3	Systematic treatment of null values	• A Null is an unknown value and every database must have a provision for storing NULL values.
# 4	Dynamic on-line catalog based on the relational model	• In addition to user data, a relational database contains data about itself, there are 2 types of tables. • User tables - that contain the 'working' data and system tables contain data about the database structure. • Metadata –that describes the structure of the database itself and includes object definitions (tables, indexes, stored procedures, etc.) and how they relate to each other. • The collection of system tables is also referred to as the system catalog or data dictionary.

Codd's 12 rules (Contd…)

VINSYS
IT SERVICES INDIA PVT. LTD.

Rule No.	Rule	Description
# 5	Comprehensive Data Sublanguage Rule	• There must be a single language that handles all communication with the database management system. • The language must support relational operations with respect to: data modification, data definition and administration.
# 6	View Updating Rule	• Database must allow for presenting data to the user in different combinations through views. • Views are nothing but virtual tables which contains extraction of data from the source tables. • Views allows to create customized snapshot of data to suit specific needs. • If a view is a simple view then it can provide update and delete operations on views.

Codd's 12 rules (Contd…)

VINSYS
IT SERVICES INDIA PVT. LTD.

Rule No.	Rule	Description
# 7	High-level insert, update and delete	• Rows are treated as sets for data manipulation operations. • A relational database should support basic operations like sub-queries, join, operators and set operations like union, intersection and minus. • Set operations and relational operators are used to operate on 'relations' (tables) to produce other relations (tables).
# 8	Physical Data Independence	• Any changes in the data storage should not affect the application that accesses it.
# 9	Logical Data Independence	• Data is logically stored in tables and physically in files. For any change in the structure of tables and relationships, the application need not be re-created.
# 10	Integrity Independence	• In order to be considered relational, data integrity must be an internal function of the DBMS; not the application program. • Data integrity means the consistency and accuracy of the data in the database (i.e., keeping the garbage out of the database).

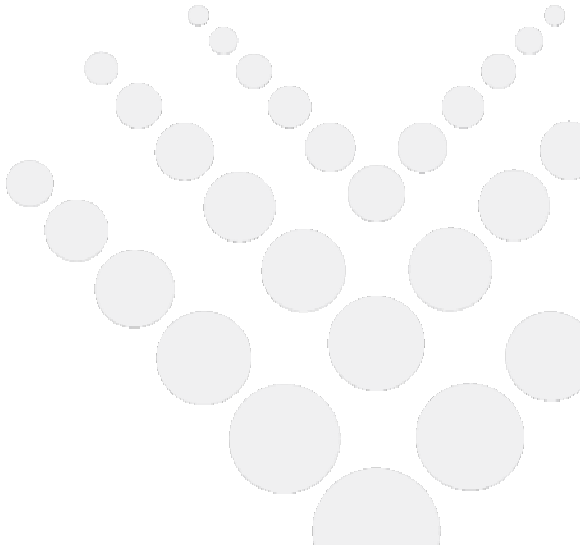
Codd's 12 rules (Contd…)

VINSYS
IT SERVICES INDIA PVT. LTD.

Rule No.	Rule	Description
# 11	Distribution Independence	• Distributed databases should support the data manipulation language of SQL. • A complex view can be created by implementing joins to extract data not only from tables on different servers (distributed queries) but also from different types of databases (heterogeneous queries). • While extracting the data from the database, the user need not be aware whether the database is distributed or not.
# 12	Non subversion Rule	• It is not possible to bypass the constraints defined on the table which means that the database integrity should never be violated.

Agenda

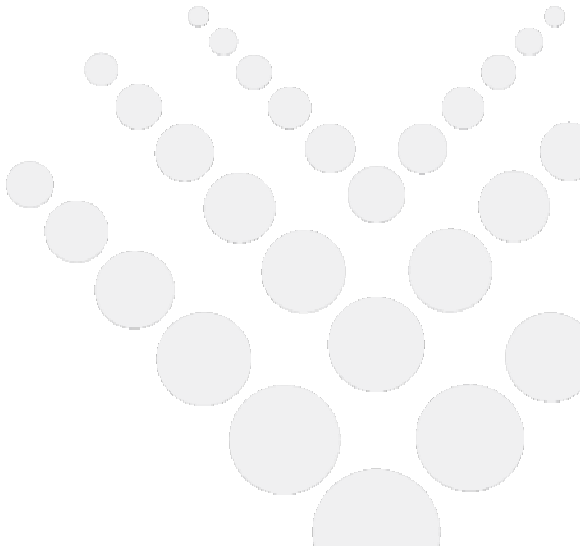
- Introduction to Entity Relationship (ER) Model.
- Entity Relationship Diagram (ERD).
- Entities, Attributes and Relationships.
- Mapping cardinalities
  - One to one.
  - One to many.
  - Many to one.
  - Many to many.
- Keys
  - Primary Key.
  - Candidate Keys.
  - Alternate Key.
  - Foreign Key.
  - Composite key.



Learning Objectives

VINSYS
IT SERVICES INDIA PVT. LTD.

- Understand the Entity-Relationship (ER) model
- Understand Entity Relationship diagram
- Understand Entities, Attributes and Relationships
- Understand different types of cardinalities
- Creating Entity Relationship Diagram
- Understand Different types of Keys



E-R Model

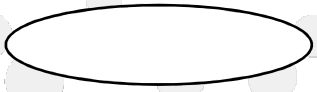
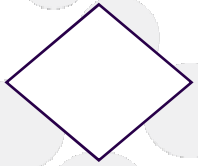
VINSYS
IT SERVICES INDIA PVT. LTD.

- **E-R model means Entity –Relationship Model.**
- It describes basic objects called entities as understood in real world and relationships among the objects
- It is a way of graphically representing the logical relationships of entities (or objects) in order to create a database
- E-R model helps to develop the logical database design of the System
- E-R model facilitates easy design of the database to represent the logical structure of the database

Entity Relationship Diagram (ERD)

VINSYS
IT SERVICES INDIA PVT. LTD.

Based on the perception of the real world, an E-R diagram is prepared as shown below

Real world	ERD	Definition	Symbol
Object	**Entity**	An entity is an any object or activity which an enterprise records.	
Property	**Attribute**	Instantiated from an object.	
Relationship	**Relationship**	Association between two dependent objects.	

Entity

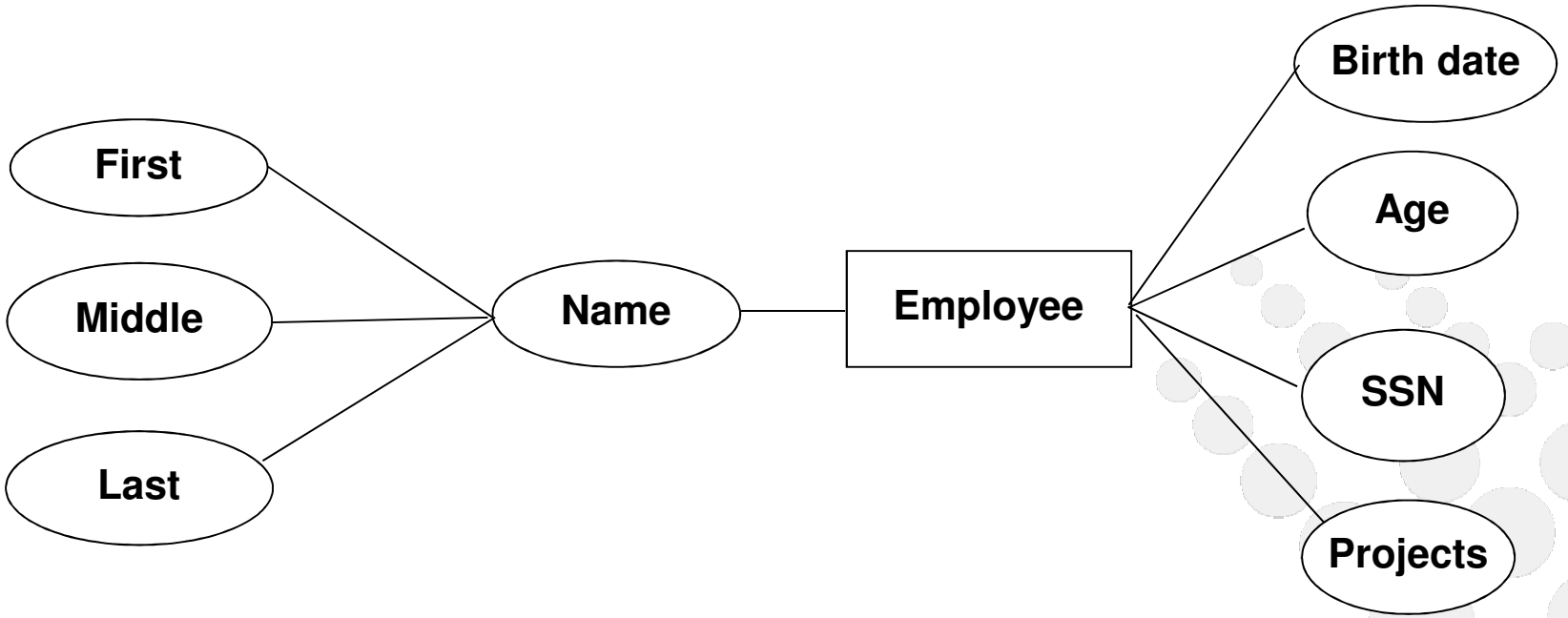
- An Entity is an object in the real world about which we want or need to maintain information.

Examples:

Persons: - information stored about people.
Ex: Customers, Employees, Authors,
Things: Companies, Assets

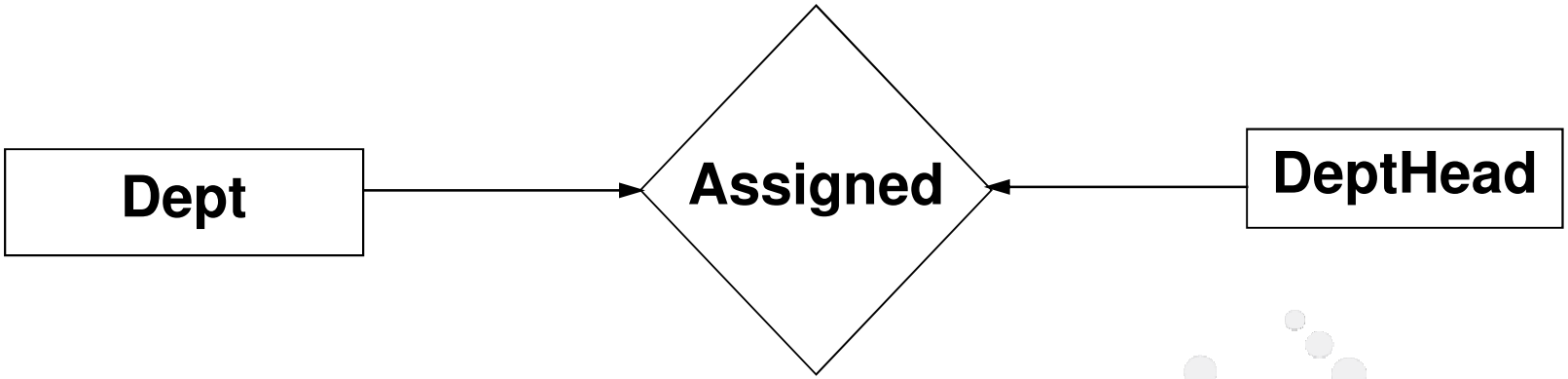
Attributes

- Attributes are the significant properties or characteristics of an entity that help in describing the entity and provide the information needed to interact with it or use it.



Relationship

Relationships are the associations between entities. They can involve one or more entities and belong to particular relationship types.



Where **Dept** and **DeptHead** are Dependent Entity.

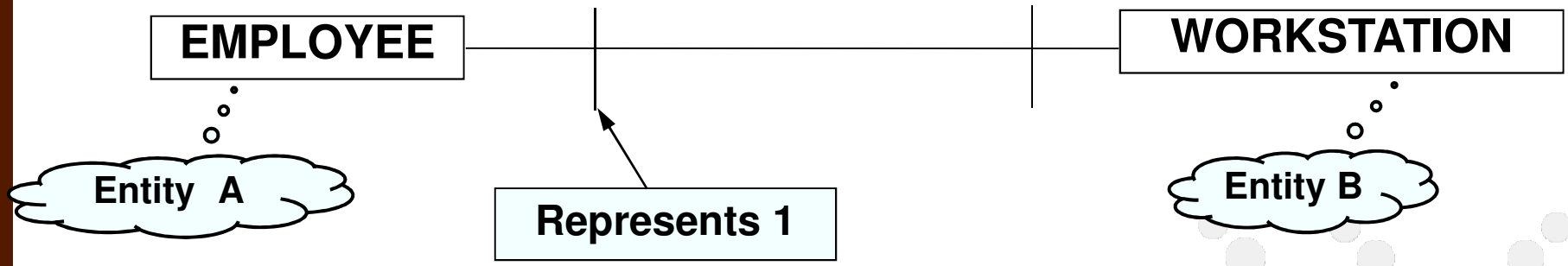
Assigned is Relationship.

Mapping cardinalities

- Cardinality is a quantity relationship between elements.
- The number of rows in the table is called as cardinality.
- Mapping cardinalities or cardinality ratio, expresses the number of entities to which another entity can be associated via a relationship set.
- The following are the different mapping cardinalities
  - One to one.
  - One to many.
  - Many to one.
  - Many to Many.

One to one (1:1)

- An entity in A is related to at most one entity in B, and an entity in B is related to at most one entity in A.
- Example 1:

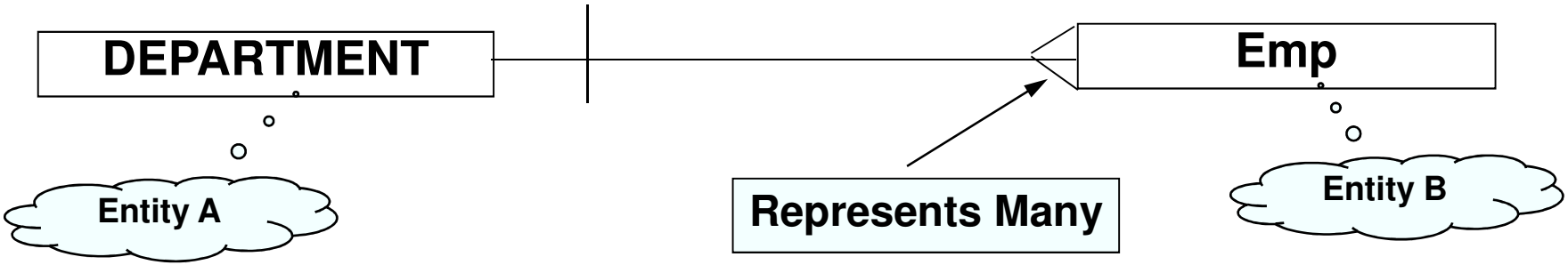


Every Employee is assigned to One Workstation.

Example 2:
One Customer can hold one Salary Account in a bank.

One to Many (1: N)

- An entity in A is related to any number of entities in B, but an entity in B is related to at most one entity in A
- Example 1:



Department consists of many Employees. But each employee will have only one Department

Example 2:
One Customer can avail different services provided by the bank

Many to One (N : 1)

- An entity in A is related to at most one entity in B, but an entity in B is related to any number of entities in A.
- Example 1:

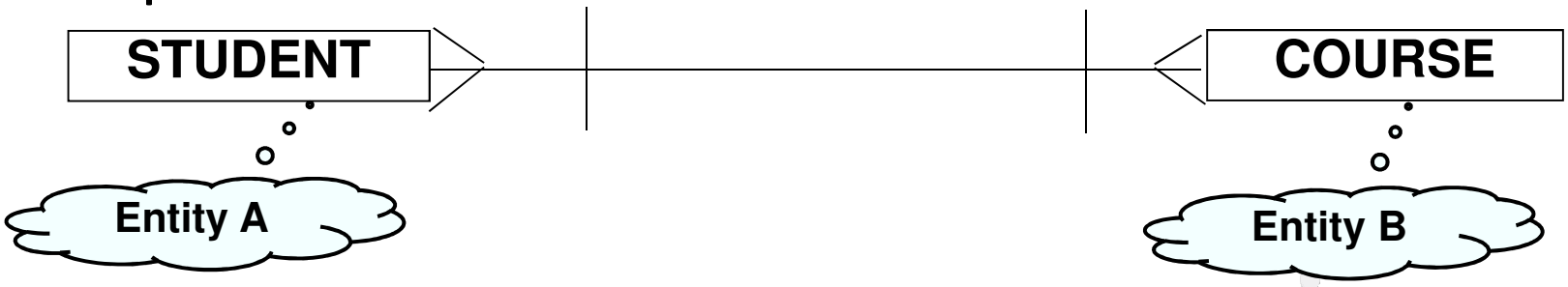
A city exists in only one state. However each state will have many cities.

Example 2:

Many Customers can hold different Accounts in one bank.

Many to Many (M:N)

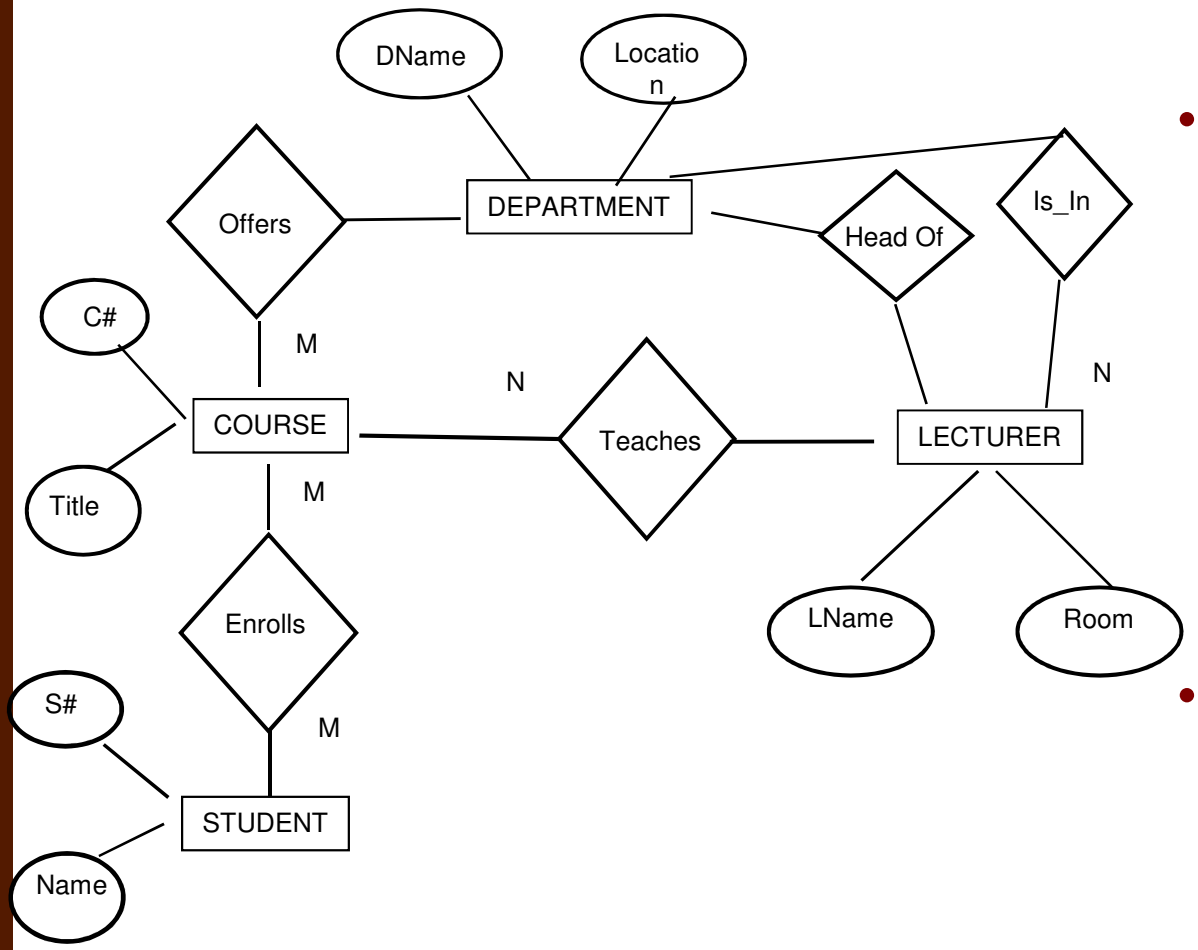
- An entity in A is related to any number of entities in B, similarly an entity in B is related to any number of entities in A
- Example 1:



A Student can enroll for more than one course. And each course will contain more than one student

Example 2:
Many products purchased by many customers

ER diagram of College DB



- There are 4 entities in this diagram
  - Course
  - Department
  - Lecturer
  - Student
- Each entity is related with another with many relationships

Keys

- Keys are used to validate the data that is being entered into a database
- **Keys play a vital role in Maintaining data accuracy in a Database**
- There are Four types of keys
  - Primary Key
  - Candidate Key
  - Alternate Key
  - Composite key

Primary Key

- **Primary key** enforces integrity of the data by uniquely identifying records. A table can have only 1 primary key

Primary Key

Emp No	Name	Designation
1000	Venkat	SSE
2000	Ramesh	Team Leader
3000	Poornima	ASE

Candidate key

- Candidate key is a Key that helps to uniquely identify row in a table
- It could potentially serve as the primary Key.

Primary Key →

Emp No	WS#	PrjName
4561	WS.A.1	Greenfield
4562	WS.B.2	Am Express
4563	WS.C.3	Avanande

Candidate Key (All rows are Unique)

The example shows the columns that either EmpNo or WS# can be identified as candidates to become a primary key since choosing any one of them will make a row unique. In these cases we have to choose voluntarily one column as primary key.

Candidate Key (Contd...)

- **Overlapping candidate key:** Two candidate keys overlap if they involve two or more attributes each (composite candidate key) and have one or more attribute in common.

Emp No	WS#	PrjName
4561	WS.A.1	Greenfield
4562	WS.B.2	Am Express
4563	WS.C.3	Avanande

First Candidate Key

Second Candidate Key

In this example you can choose EmpNo and WS# to unique identify the row or WS# and PrjName, in which the WS# is common.

Alternate Key

- **Non-key attribute:** Attribute that is not identified as a candidate key
- **Alternate Key:** Any of the candidate keys that is not the primary key is called an alternate key

Foreign Key

- **Foreign key :**It provides relationship between table in the database
- In relational database, a foreign key of a table is a set of Column/Columns that references the primary key of another table
- The enforcement of this Constraint is Known as **Referential Integrity**

Composite Key

- A key composed of more then one column is called composite key or concatenated Key

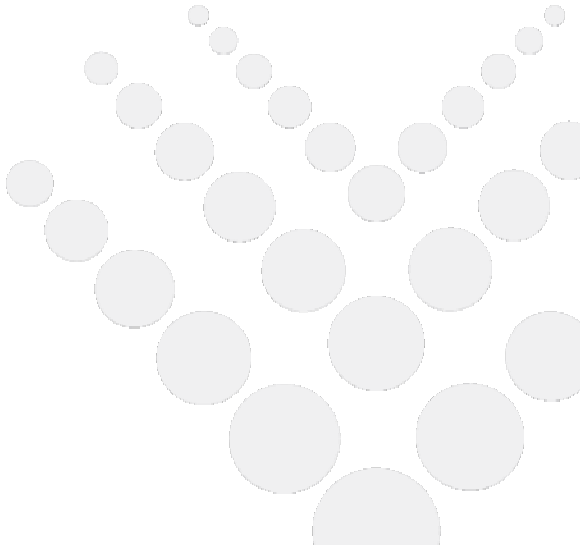
EmpNo	WS#	Start date	End date	Hours single
4561	AS12	12/11/2005	12/3/2006	150
4562	AS13	12/11/2005	12/11/2005	8
4563	AS14	13/11/2005	13/11/2005	8

Primary key is a combination of Emp No. and WS#

Agenda

VINSYS
IT SERVICES INDIA PVT. LTD.

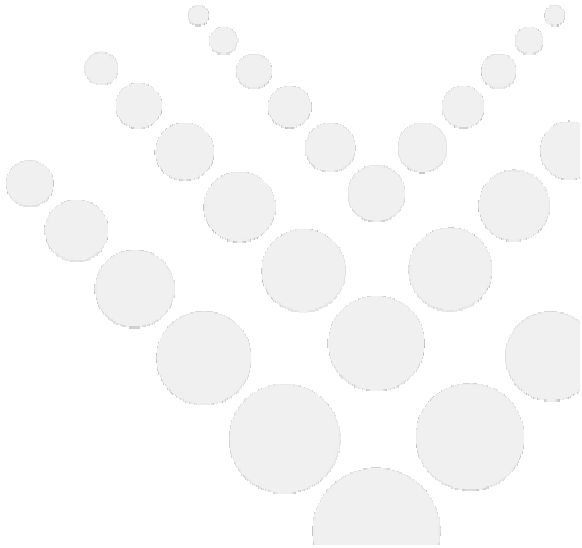
- Normalization.
- Types of Normal forms.
- First Normal Form.
- Functional Dependency.
- Full Functional Dependence and Partial Functional Dependence.
- Second Normal Form.
- Transitive Dependency.
- Third Normal Form.
- Denormalization.



Agenda

VINSYS
IT SERVICES INDIA PVT. LTD.

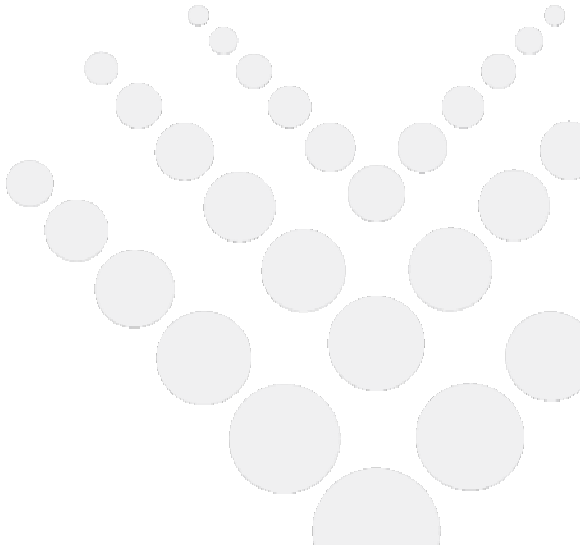
- Normalization.
- Types of Normal forms.
- First Normal Form.
- Functional Dependency.
- Full Functional Dependence and Partial Functional Dependence.
- Second Normal Form.
- Transitive Dependency.
- Third Normal Form.
- Denormalization.



Learning Objectives

VINSYS
IT SERVICES INDIA PVT. LTD.

- Understand Database design techniques.
- Understand the need for normalization.
- Understand different types of normalization.
- Understand Functional dependencies.
- Understand Denormalization.



Normalization

VINSYS
IT SERVICES INDIA PVT. LTD.

- **Normalization** is the process of efficiently organizing data in a database.
- It is a design technique that is widely used as a guide in designing relational databases.
- Normalization is essentially a two step process
  - It puts data into tabular form by removing repeating groups.
  - It removes duplicated data from the relational tables.
- Normalization theory is based on the concepts of normal forms.

Normal Forms

VINSYS
IT SERVICES INDIA PVT. LTD.

- A relational table is said to be in a normal form if it satisfies a certain set of constraints.
- They are special properties and constraints that a table schema should possess in order to achieve certain desired goals like minimizing redundancy of data.
- There are six normal forms that have been defined

❑ First Normal Form (1NF)	❑ Second Normal Form (2NF)
❑ Third Normal Form (3NF)	❑ Boyce Codd Normal Form (BCNF)
❑ Fourth Normal Form (4NF)	❑ Fifth Normal Form (5NF)

The Third Normal Form is quite sufficient for most business database design purposes.

Example of Normalization

VINSYS
IT SERVICES INDIA PVT. LTD.

- Normalize the following Data

item_no	item_desc	price	manufacturer_id	manufacturer_name	distributor_id	distributor_name	order_id	date_entered	Qty
1	Watches	3000	M1001	Aqua Marine	D1001	Marine World	10010 10011	12-May-07	20
2	Pens	1000	M1002	Magnum	D1002	Water World	10011 10012	14-May-07	10
3	Mobiles	4000	M1003	Coral	D1003	Alpha Company	10010 10012	16-May-07	15

First Normal Form

VINSYS
IT SERVICES INDIA PVT. LTD.

- A design is said to be in first normal form(1NF) when there are no repeating groups and all attributes are dependent on the primary key.

Example:

In a table the Address field can be further divided into smaller segment like:

House Number,
Street Name1,
City,
PIN etc.

So, in a table if we have a column called "Address" we can not consider it in **first Normal Form**.

First Normal Form (Contd…)

VINSYS
IT SERVICES INDIA PVT. LTD.

item_no	item_desc	price	manufacturer_id	manufacturer_name	distributor_id	distributor_name	order_id	date_entered	Qty
1	Watches	3000	M1001	Aqua Marine	D1001	Marine World	10010	12-May-07	20
1	Watches	3000	M1001	Aqua Marine	D1001	Marine World	10011	12-May-07	20
2	Pens	1000	M1002	Magnum	D1002	Water World	10011	14-May-07	10
2	Pens	1000	M1002	Magnum	D1002	Water World	10012	14-May-07	10
3	Mobiles	4000	M1003	Coral	D1003	Alpha Company	10010	16-May-07	15
3	Mobiles	4000	M1003	Coral	D1003	Alpha Company	10012	16-May-07	15

Functional Dependency

- A functional dependency exists when the value of one item is fully determined by another.
- Example:
  - given the relation Student(StudID, StudName, Marks), attribute StudName is functionally dependant on attribute StudID.
  - This is written as **StudID → StudName.**
- In other words attribute A is functionally dependent on B if and only if for each value of B, there is exactly one Value of A. Attribute B is called determinant.
- For any given value for attribute A, there is just one corresponding value of attribute B.

Functional Dependency(2 of 2)

VINSYS
IT SERVICES INDIA PVT. LTD.

- A functional dependency exists when the value of one item is fully determined by another.
- Example:
  - given the relation Student(StudID, StudName, Marks), attribute StudName is functionally dependant on attribute StudID.
  - This is written as **StudID → StudName.**
- In other words attribute A is functionally dependent on B if and only if for each value of B, there is exactly one Value of A. Attribute B is called determinant.
- For any given value for attribute A, there is just one corresponding value of attribute B.

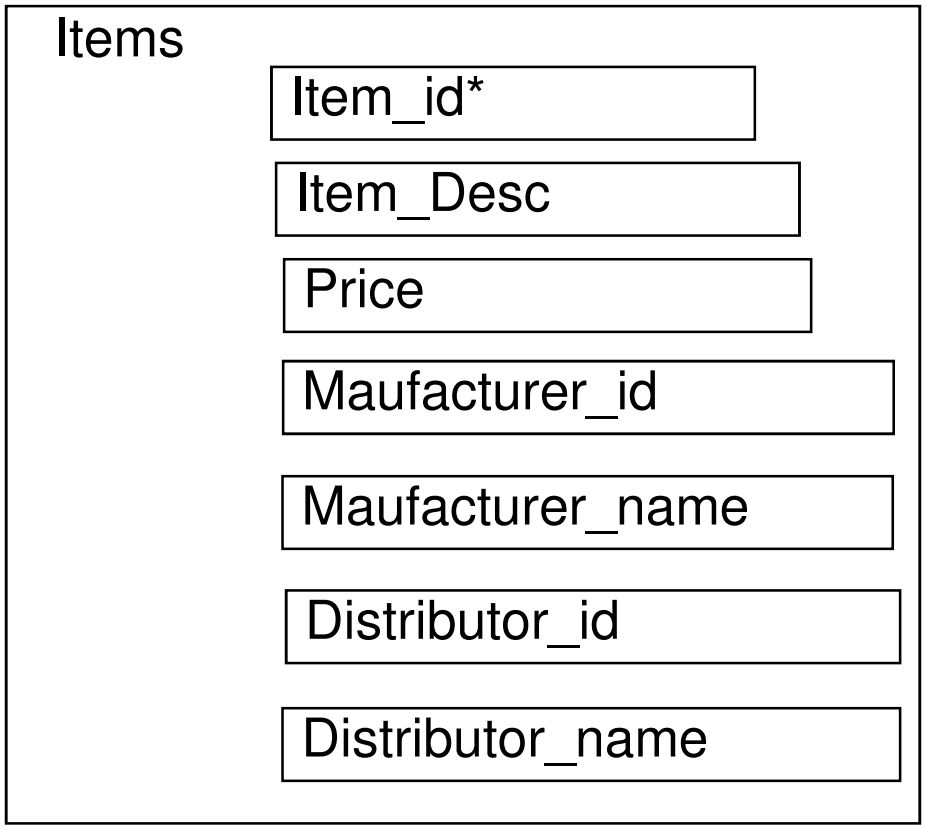
Example of Functional Dependency

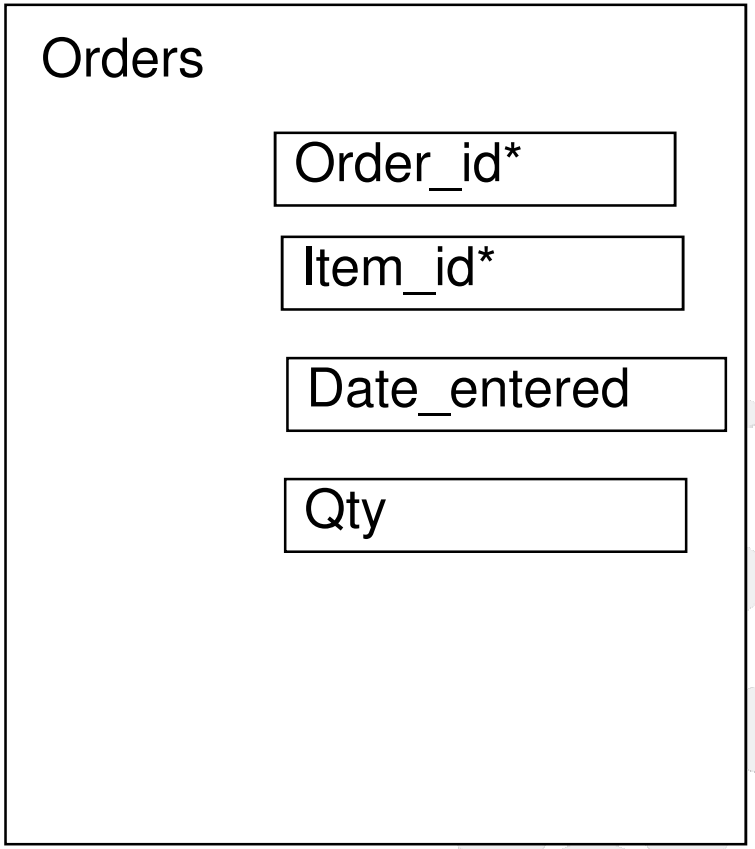
Code	Name	City
E1	Richard	CA
E2	Thomson	Delhi

- Given a particular value of *Code* there is precisely one corresponding value for *Name*
  - Example:
    - For code **E1** there exactly one value of name **Richard**.
    - **Name** is functionally dependent on **Code.**

Second Normal Form

- An entity type is in second normal form (2NF) if it is in 1NF and all its attributes are dependent on the primary key. In other words, there are no attributes that are dependent on either none of or only part of the primary key.





Transitive Dependency

- A **transitive dependency** is a type of functional dependency in which the value in a non-key field is determined by the value in another non-key field and that field is not a candidate key.

* ProjectNum	ProjectTitle	ProjectMgr	Phone
30-452-T3	STAR manual	Garrison	2756
30-457-T3	ISO procedures	Jacanda	2954
30-482-TC	Web site	Friedman	2846
31-124-T3	Employee handbook	Jones	3102
31-238-TC	STAR prototype	Garrison	2756
31-241-TC	New catalog	Jones	3102
35-152-TC	STAR pricing	Vance	3022
36-272-TC	Order system	Jacanda	2954

The phone number is dependent on the manager, which is dependent on the project number (a transitive dependency)

The ProjectMgr field is not a candidate key because the same person manages more than one project

Third Normal Form

An entity is in third normal form (3NF) if it is in 2NF and all of its attribute are dependent on only the primary key.

In other words, all non-key attributes are mutually independent and are fully dependent on the primary key.

To make sure that an entity is in 3rd Normal form, we need to remove any column which is depending on any other attribute of the table.

Example of 3NF

- Example : Let us assume that we have the following structure for a table:

Column Name	Constraint
Book_ID	Primary Key
Book_Name	
Publisher_ID	
Publisher_Name	
Publisher_Address	
ISBN_Number	

Example of 3NF (Contd…)

- In the above example, Publisher_Address column is not depending on the Book_ID (which is the Primary Key), rather it depends on Publisher_Name. So, to make the entity in 3rd Normal Form, we need to restructure the table as follows:

Book Table	
Column Name	**Constraint**
Book_ID	Primary Key
Book_Name	
Publisher_ID	Foreign Key
ISBN_Number	

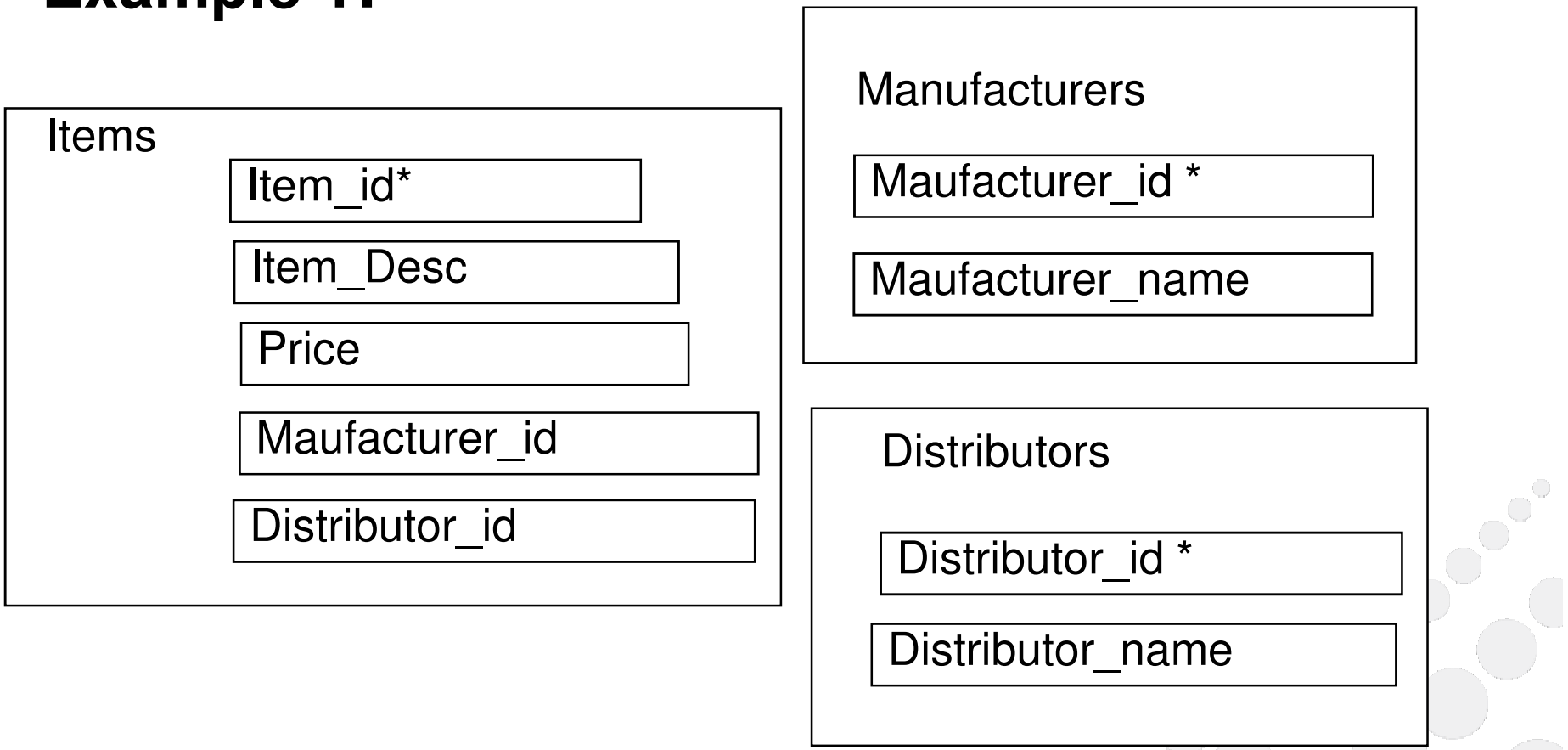
We shall have another table which will hold the data about the Publishers:

Publisher Table	
Column Name	**Constraint**
Publisher_ID	Primary Key
Publisher_Name	
Publisher_Address	

Example of 3NF

VINSYS
IT SERVICES INDIA PVT. LTD.

Example 1:

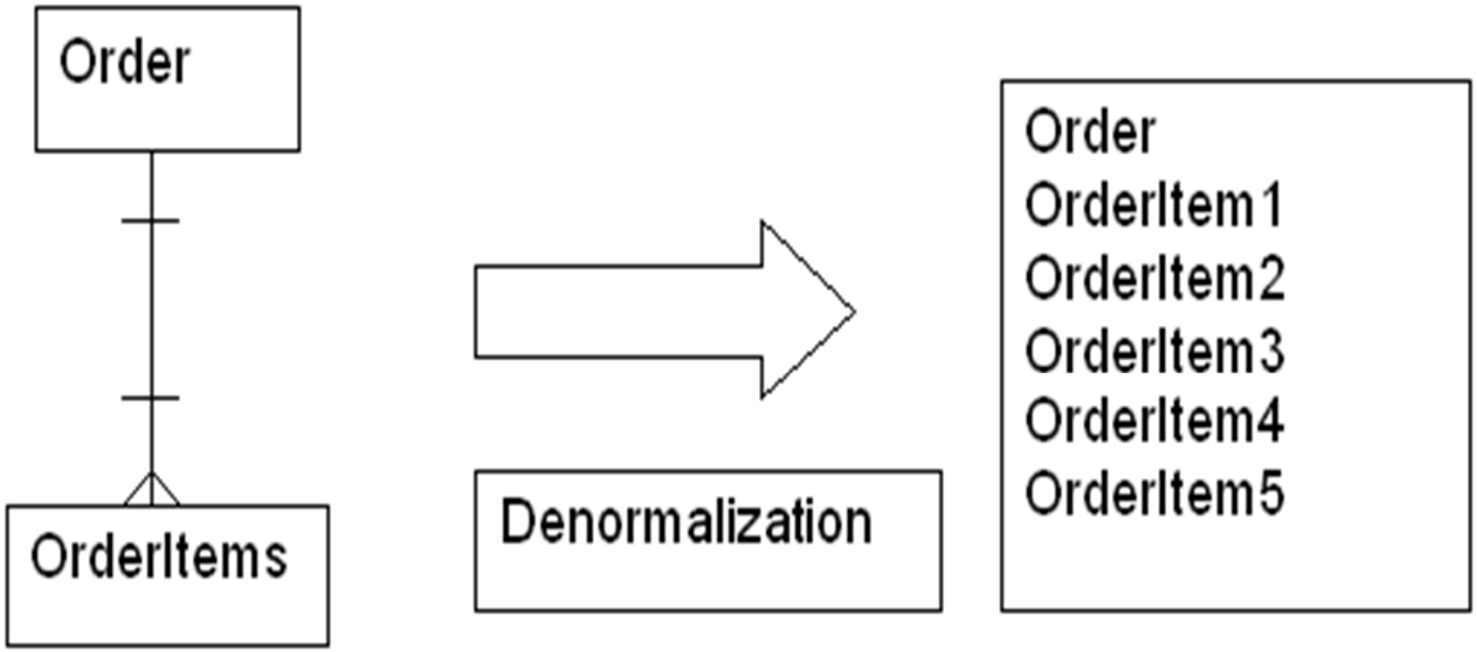


DENORMALIZATION

VINSYS
IT SERVICES INDIA PVT. LTD.

- **What is DENORMALIZATION?**
- Denormalization is a technique which is the opposite of normalization. In order to speed up the database access, it suggests to move from higher level of normalization to lower normal forms of database modeling.
- **NEED FOR DENORMALIZATION**
- Consider 2 tables –Orders and OrderItems which stores the order and items to be ordered respectively. As per your analysis, you have created a view called Order_Item_View. This is one of the most heavily used views in the database. To read data from Order and the OrderItems table will require a join statement and access to many physical pages. To write data will require several update statements. Moreover 90 percent of Orders have no more than 5 positions. To avoid joins, you can merge the two tables into 1 table.
- Data is normalized to avoid maximum redundancy. They are not optimized for minimum access time. In the case of denormalization process, time does not play a role. The only reason why denormalization is done is to enhance the performance.

Denormalization (Contd…)



3: Database Design Techniques
Key Points

- **Normalization** is the process of efficiently organizing data in a database.
- Normalization is designed to logically address potential problems with information stored in a database.
- There are six normal forms.
- Typically a database will be in Third Normal Form.

3: Database Design Techniques
Key Points (Contd…)

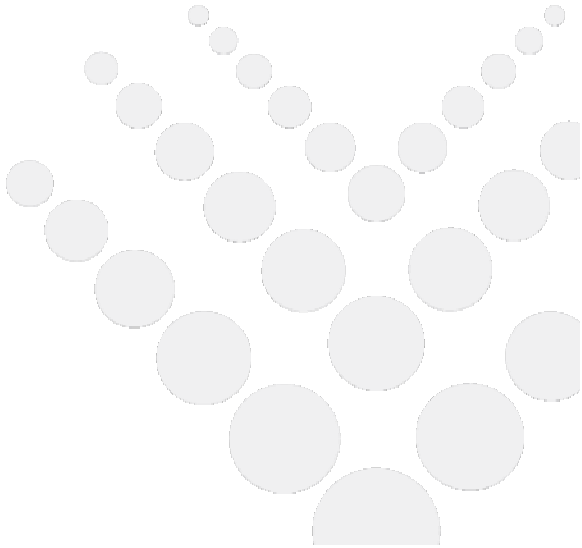
VINSYS
IT SERVICES INDIA PVT. LTD.

- A design is said to be in first normal form(1NF) when there are no repeating groups and all attributes are dependent on the primary key.
- An entity type is in second normal form (2NF) if it is in 1NF and all its attributes are dependent on the primary key.
- A **transitive dependency** is a type of functional dependency in which the value in a non-key field is determined by the value in another non-key field and that field is not a candidate key.

Agenda

VINSYS
IT SERVICES INDIA PVT. LTD.

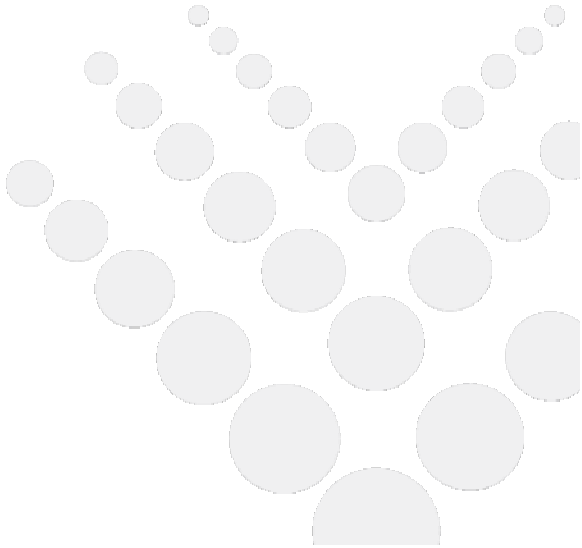
- What is SQL.
- The SQL sublanguages.
- Data Definition Language.
- Data Manipulation Language.
- Data Control Language Transaction Control.



Objectives

VINSYS
IT SERVICES INDIA PVT. LTD.

- Understand SQL.
- Understand the Sublanguages of SQL.
- Understand basic SQL queries.



Structured Query Language (SQL)

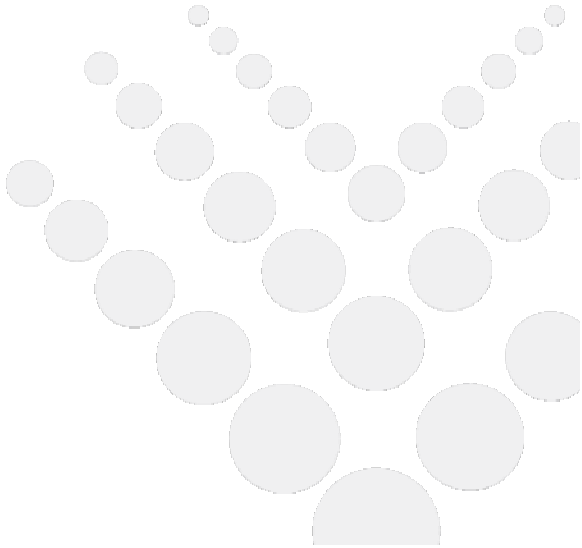
VINSYS
IT SERVICES INDIA PVT. LTD.

- SQL is **Structured Query Language**.
- It is a special-purpose, nonprocedural language that supports the definition, manipulation, and control of data in relational database management systems.
- SQL is used to manipulate and retrieve data stored in a database.
- SQL is the most commonly used query language available.
- SQL was designed by IBM during 1970's.
- Word **SQL** is derived from **"Sequel".**

Structured Query Language (SQL) (Contd…)

VINSYS
IT SERVICES INDIA PVT. LTD.

- Depending on the functionality SQL is classified as follows:
  - Data Definition Language (DDL)
  - Data modification Language (DML)
  - Data Control Language (DCL)
  - Transaction Control Language (TCL)



Sub-languages of SQL

VINSYS
IT SERVICES INDIA PVT. LTD.

SQL is segregated into DDL,DML, DCL and TCL.
Data retrieval : SELECT

DDL	DML	DCL	TCL
CREATE	INSERT	GRANT	COMMIT
ALTER	UPDATE	REVOKE	ROLLBACK
DROP	DELETE		SAVEPOINT
TRUNCATE			

JOINS

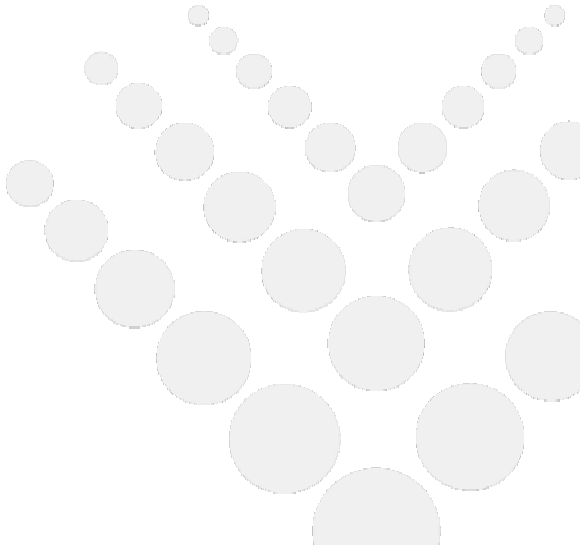
VINSYS
IT SERVICES INDIA PVT. LTD.

Joins are normally used to retrieve data from more than one table.

The JOIN keyword joins one or more tables together in a results set

To perform a join there must be a common key that defines how the rows in the table correspond to each other.

All subqueries can be converted to joins but all joins cannot be converted to subqueries.



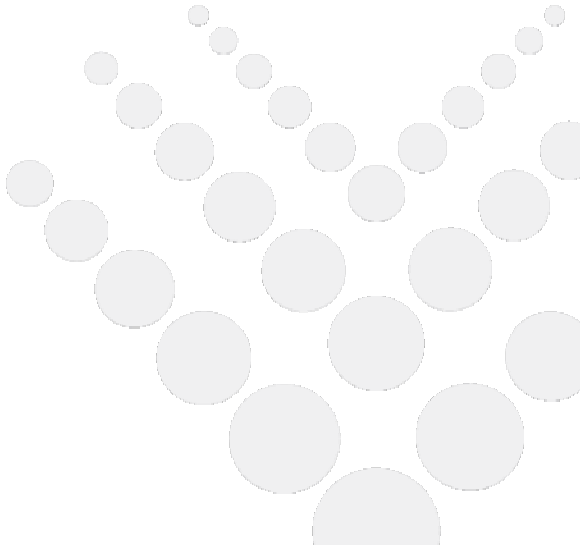
Types of Joins

Different types of joins are

- Inner join
- Outer join

  Left-outer join

  Right-outer join
  Full outer join
- Self join

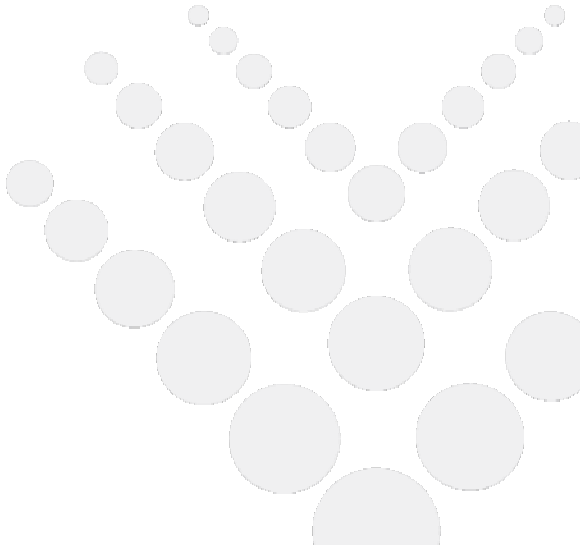


Inner Join

VINSYS
IT SERVICES INDIA PVT. LTD.

The inner join joins two or more tables returning only matched rows

Inner Join requires that both tables involved in the join must have a primary-foreign key relationship



Inner Join (contd...)

VINSYS
IT SERVICES INDIA PVT. LTD.

Example: To display the ename, job, sal and dname in which the employees are working

```
SELECT ENAME,
JOB,SAL, DNAME
FROM DEPT D,EMP E
WHERE
D.DEPTNO=E.DEPTNO;
```

Sample Output →

ENAME	JOB	SAL	DNAME
SMITH	CLERK	800	RESEARCH
ALLEN	SALESMAN	1600	SALES
WARD	SALESMAN	1250	SALES
JONES	MANAGER	2975	RESEARCH
MARTIN	SALESMAN	1250	SALES
BLAKE	MANAGER	2850	SALES
CLARK	MANAGER	2450	ACCOUNTING
SCOTT	ANALYST	3000	RESEARCH
KING	PRESIDENT	5000	ACCOUNTING
TURNER	SALESMAN	1500	SALES
ADAMS	CLERK	1100	RESEARCH
JAMES	CLERK	950	SALES
FORD	ANALYST	3000	RESEARCH
MILLER	CLERK	1300	ACCOUNTING

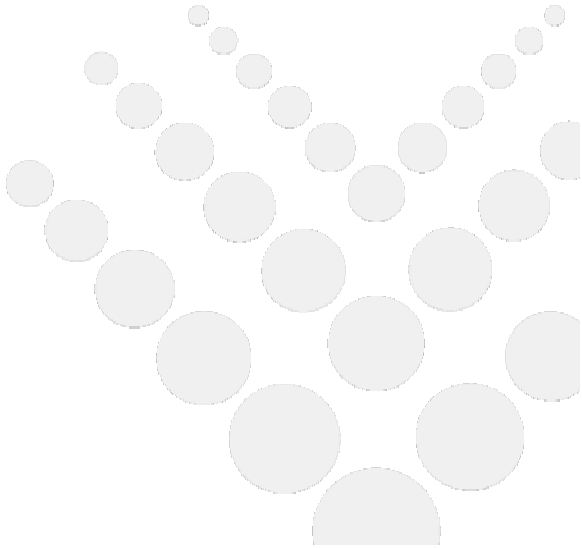
Outer Join

Outer Join returns not only the common rows from the tables but also returns the rows that are unique in the tables

In Outer Join, rows are returned even if the rows from one table is not matching with those of another table

There are three types of Outer Joins

- Left outer join
- Right outer join
- Full outer join



Left Outer Join

In left outer join, all the records from the table on the left of the OUTER JOIN statement are returned

Syntax

```
SELECT column_list FROM left_table LEFT
[OUTER] JOIN right_table ON condition
```

```
SELECT column_list FROM table t1,table t2
where t1.column_name=t2.column_name (+)
```

Left Outer Join (contd…)

Example: To display the ename, job, sal and dname in which the employees are working along with the dname in which no employees are working.

```
SELECT ENAME,
JOB,SAL, DNAME
FROM DEPT D
LEFT JOIN  EMP E
ON D.DEPTNO =
E.DEPTNO;
```

Sample Output →

```
ENAME          JOB       SAL       DNAME
=====================================
SMITH          CLERK      800    RESEARCH
ALLEN      SALESMAN      1600       SALES
WARD       SALESMAN      1250       SALES
JONES       MANAGER      2975    RESEARCH
MARTIN     SALESMAN      1250       SALES
BLAKE       MANAGER      2850       SALES
CLARK       MANAGER      2450  ACCOUNTING
SCOTT       ANALYST      3000    RESEARCH
KING      PRESIDENT      5000  ACCOUNTING
TURNER     SALESMAN      1500       SALES
ADAMS          CLERK      1100    RESEARCH
JAMES          CLERK       950       SALES
FORD        ANALYST      3000    RESEARCH
MILLER         CLERK      1300  ACCOUNTING
                                    OPERATIONS
```

Right Outer Join

In right outer join, all the records from the table on the right of the OUTER JOIN are returned

Syntax

```
SELECT column_list FROM left_table RIGHT
[OUTER] JOIN right_table ON condition
```

```
SELECT column_list FROM table
t1,table t2 where t1.coluumn_name
(+) =t2.column_name
```

Right Outer Join (contd…)

VINSYS
IT SERVICES INDIA PVT. LTD.

Example: To display the ename, job, sal and dname in which the employees are working along with the dname in which no employees are working.

```
SELECT ENAME,
JOB,SAL, DNAME
FROM EMP E
RIGHT JOIN  DEPT D
ON E.DEPTNO =
D.DEPTNO;
```

Sample Output →

```
ENAME          JOB       SAL       DNAME
=====================================
SMITH          CLERK      800    RESEARCH
ALLEN      SALESMAN      1600       SALES
WARD       SALESMAN      1250       SALES
JONES       MANAGER      2975    RESEARCH
MARTIN     SALESMAN      1250       SALES
BLAKE       MANAGER      2850       SALES
CLARK       MANAGER      2450  ACCOUNTING
SCOTT        ANALYST      3000    RESEARCH
KING       PRESIDENT      5000  ACCOUNTING
TURNER     SALESMAN      1500       SALES
ADAMS          CLERK      1100    RESEARCH
JAMES          CLERK       950       SALES
FORD         ANALYST      3000    RESEARCH
MILLER         CLERK      1300  ACCOUNTING
                                      OPERATIONS
```

Full Outer Join

A full outer join is essentially a combination of left and right outer joins i.e..,

- The records from the table on left are included even if there are no matching records on the right
- The records from the table on the right are included even if there are no matching records on the left.

Syntax:

```
SELECT column_list FROM left_table FULL
[OUTER] JOIN right_table ON condition
```

Full Outer Join (contd...)

VINSYS
IT SERVICES INDIA PVT. LTD.

Example:

```
SELECT ENAME,
JOB,SAL, DNAME
FROM EMP E FULL
JOIN  DEPT D
ON E.DEPTNO =
D.DEPTNO;
```

Sample Output :

ENAME	JOB	SAL	DNAME
SMITH	CLERK	800	RESEARCH
ALLEN	SALESMAN	1600	SALES
WARD	SALESMAN	1250	SALES
JONES	MANAGER	2975	RESEARCH
MARTIN	SALESMAN	1250	SALES
BLAKE	MANAGER	2850	SALES
CLARK	MANAGER	2450	ACCOUNTING
SCOTT	ANALYST	3000	RESEARCH
KING	PRESIDENT	5000	ACCOUNTING
TURNER	SALESMAN	1500	SALES
ADAMS	CLERK	1100	RESEARCH
JAMES	CLERK	950	SALES
FORD	ANALYST	3000	RESEARCH
MILLER	CLERK	1300	ACCOUNTING
			OPERATIONS

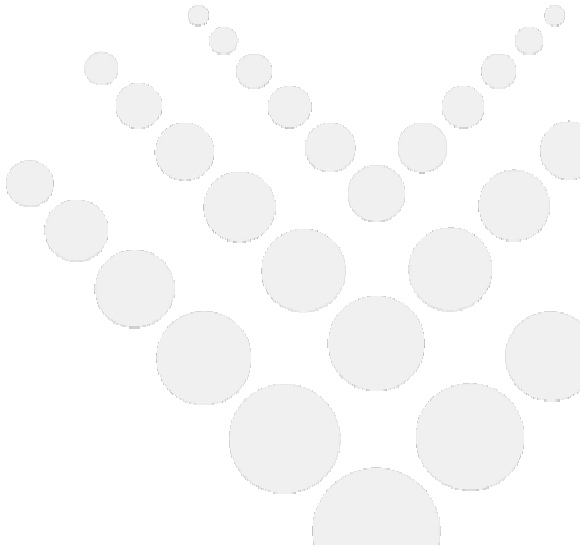
Self Join

VINSYS
IT SERVICES INDIA PVT. LTD.

A self-join is a query in which a table is joined (compared) to itself.

It is used to compare values in a column with other values in the same column in the same table.

Self-joins are also very useful in conjunction with sub queries.



Introduction to Views

VINSYS
IT SERVICES INDIA PVT. LTD.

- Views are virtual tables (tables which don't occupy memory) whose contents are taken or derived from other tables.
- Just like other tables, a view consists of rows with columns, and you can retrieve data from a view.
- Views can be used to join two tables in database and present the data as if the data were coming from a single table
- Views can be used as a security mechanism to restrict the data available to end users
- Views can also be used to aggregate data.

Example of a View

EmpNo	Ename	Empsal	Empaddr	Empcity
101	Jack	10000	jonvis	London
102	James	20000	kali	Paris
103	Steve	30000	Havens	Newyork
104	Peter	40000	Lawrence	Paris
105	Krik	50000	cham	London

This example shows creating a view called EmpView from the EMP table.

This view contains two columns EmpNo and Empname which are taken from EMP table. The actual data will be in the EMP table, which can be manipulated from the EmpView.

EmpNo	Empname
101	jack
102	james
103	steve

Creating a VIEW

VINSYS
IT SERVICES INDIA PVT. LTD.

Syntax:

```
CREATE VIEW viewname AS <select_statement>
```

Example to create a view from Supplier table

```
CREATE VIEW v_supp
AS SELECT  Supp_no,Supp_name,supp_city FROM
Supplier;
```

Example to create a view from Supplier table for City in MUM

```
CREATE VIEW v_supp_City
AS SELECT  supp_no, supp_name, supp_city FROM
Supplier S WHERE City = 'MUM';
```

Simple view

VINSYS
IT SERVICES INDIA PVT. LTD.

Simple View is also known as normal view.
It contains data only from one table.
It wont contain JOINS or GROUP By statements.
DML operations can be performed on this view.

Example

```
CREATE VIEW v_supp
AS SELECT supp_no, supp_name FROM Supplier;
```

The example creates a simple view v_supp which actually selects supp_no and supp_name from Supplier Table.

Complex View

VINSYS
IT SERVICES INDIA PVT. LTD.

It will be accessed from multiple tables
Normally a complex view will contain joins, sub queries and aggregate functions.
It contains group functions to display values from two tables.
DML operations cannot be performed through this view.

Example

```
CREATE VIEW v_dept_emp(dname, job, maxSal, minSal, avgSal)
AS  SELECT dname, job, max(sal), min(sal), avg(sal)FROM  dept
d, emp e WHERE  d.deptno = e.deptno GROUP BY dname, job;
```

The example creates a complex view v_dept_emp which selects the dname column, maximum sal, minimum sal and average sal. It uses three Aggregate functions and it joins two tables Dept and Emp. The column names of the view are mentioned in the create view statement.

Read Only View

With **read only** clause with the view definition does not allow any DML operation on that view.

The view is read only (I.e the view can only be queried)

Example

```
CREATE VIEW v_emp_job AS
SELECT  empno,ename,job FROM emp
With Read only;
```

The example creates a simple view using the with read only clause which selects empno, ename and job from Emp Table . No DML operations can be carried through this view.

Subqueries

VINSYS
IT SERVICES INDIA PVT. LTD.

Subqueries are also called as Nested queries which means SELECT statement can be nested inside another SELECT statement.

The subquery is placed in the WHERE clause of a SELECT statement.

It allows to group multiple SELECT statements together.

There are 2 types of Sub queries:

- Independent/Non-correlated sub queries and
- Correlated sub queries

Subqueries can return either a single row or multiple rows.

Independent/Non-Correlated Sub queries

A subquery is also called as an inner query.

The query which is placed before the inner query is called as the Outer query or Parent Query

In Independent Sub queries

- Inner query is executed independent of outer query.
- The inner query executes only once.
- Inner query is executed first and the results replace the inner SELECT statement.
- Outer query then executes based on the results provided by the inner query.

Sequences

VINSYS
IT SERVICES INDIA PVT. LTD.

- **A SEQUENCE…**
  - Implicitly generates **sequential values.**
  - Can effectively be used for **auto-generation of Primary Key** Values.
  - Speeds up the efficiency of accessing sequence values when cached in memory.
  - Is a **sharable** Object.
  - Requires extremely careful handling
    - **As gaps in sequence may occur because of rollback / deletion or sharability**

Creation of Sequence

VINSYS
IT SERVICES INDIA PVT. LTD.

```
CREATE SEQUENCE sequence_name

                [START WITH n]

                [INCREMENT BY n]

                [{MAXVALUE n | NOMAXVALUE}]

                [{MINVALUE n | NOMINVALUE}]

                [{CYCLE | NOCYCLE}]

                [{CACHE n | NOCACHE}] ;
```

Use of Sequence

```
CREATE SEQUENCE Seq1
        START WITH 1
        INCREMENT BY 2
        MAXVALUE 5
        NOCACHE
        NOCYCLE;
```

```
CREATE SEQUENCE Seq2
        START WITH 1
        INCREMENT BY 2
        MAXVALUE 5
        NOCACHE
        CYCLE
```

We can check the details of sequences created

```
SELECT *  FROM user_sequences;
```