Government Arts College, Coimbatore(Autonomous)

PG and Research Department of Information Technology

III B.Sc(IT)- RELATIONAL DATABASE MANAGEMENT SYSTEM

Sub Code:18BIT52C

Prepared by Dr.P.Radha

SYLLABUS

UNIT I: Database Concepts: A Relational approach: Database–Relationships–DBMS–Relational Data Model–Integrity Rules–Theoretical Relational Languages. Database Design: Data Modeling and Normalization: Data Modeling–Dependency–Database Design – Normal forms– Dependency Diagrams -Renormalizations – Another Example of Normalization.

UNIT II: Oracle9i: Overview: Personal Databases–Client/ServerDatabases–Oracle9i an introduction– SQL*Plus Environment–SQL–Logging into SQL*Plus–SQL*Plus Commands–Errors &Help–Alternate Text Editors-SQL*Plus Worksheet – iSQL*Plus .Oracle Tables: DDL: Naming Rules and conventions– Data Types–Constraints–Creating Oracle Table–Displaying Table Information–Altering an Existing Table–Dropping, Renaming, Truncating Table–Table Types–Spooling–Error codes.

UNIT III: Working with Tables: Data Management and Retrieval: DML – Adding a new Row/Record – Customized Prompts – Updating and Deleting Existing Rows/Records–Retrieving Data from A Table– Arithmetic Operations– Restricting Data with WHERE clause– Sorting– Revisiting Substitution Variables–DEFINE command–CASE structure. Functions and Grouping: Built-in functions – Grouping Data Multiple Tables: Joins and Set operations: Join–Set Operators.

UNIT IV: PL/SQL: A Programming Language: History–Fundamentals–Block Structure–Comments – Data Types–Other Data Types–Variable Declaration–Anchored Declaration-Assignment operation – Bind Variables–Substitution Variables–Printing–Arithmetic Operators. Control Structures and Embedded SQL: Control Structures–Nested Blocks–SQL in PL/SQL–Data Manipulation –Transaction Control statements. PL/SQL Cursors and Exceptions: Cursors – Implicit & Explicit Cursors and Attributes–Cursor FOR loops–SELECT…FOR UPDATE – WHERE CURRENT OF clause– Cursor with Parameters–Cursor Variables– Exceptions– Types of Exceptions.

UNIT V: PL/SQL:PL/SQL Composite Data Types: Records–Tables– arrays. Named Blocks: Procedures–Functions–Packages–Triggers – Data Dictionary Views.

UNIT I

Database Concepts: A Relational approach: Database–Relationships–DBMS–Relational Data Model–Integrity Rules–Theoretical Relational Languages. Database Design: Data Modeling and Normalization: Data Modeling–Dependency–Database Design – Normal forms– Dependency Diagrams - Renormalizations – Another Example of Normalization.

Text Book: “DATABASE SYSTEMS USING ORACLE”-NILESH SHAH,2nd Edition, PHI

Prepared by Dr.P.Radha

Database Concepts: A Relational Approach

- Database
- Relationships
- DBMS
- Relational Data Model
- Integrity Rules
- Theoretical Relational Languages

DATABASE AN INTRODUCTION

- A database is an electronic store of data
- It is a repository that stores information about different "things" and also contains relationships among those different "things"

CONT..

Let us examine some of the basic terms used to describe the structure of a database:

- A person, place, event, or item is called an **entity**
- The facts describing an entity are known as **data.**
- For example, if you were a registrar in a college, you would like to have all the information about the students. Each student is an entity in such a scenario

CONT..

- Each entity can be described by its characteristics, which are known as attributes.
- For example, some of the likely attributes for a college student are student identification number, last name, first name, phone number, Social Security number, gender, birthdate, and so on.

CONT..

- All the related entities are collected together to form an entity set.
- An entity set is given a singular name. For example, the STUDENT entity set contains data about students only.
- All related entities in the STUDENT entity set are students. Similarly, a company keeps track of all its employees in an entity set called EMPLOYEE.
- The EMPLOYEE entity set does not contain information about the company's customers, because it wouldn't make any sense.

CONT..

- A database is a collection of entity sets.
- For example, a college's database may include information about entities such as student, faculty, course, term, course section, building, registration information, and so on.

CONT..

- The entities in a database are likely to interact with other entities. The interactions between the entity sets are called relationships.
- The interactions are described using active verbs.
- For example, a student takes a course section (CRSSECTION), so the relationship between STUDENT and CRSSEC¬TION is takes.
- A faculty member teaches in a building, so the relationship between FACULTY and BUILDING is teaches

RELATIONSHIPS

- The database design requires you to create entity sets, each describing a set of related entities.
- The design also requires you to establish all the relationships between the entity sets within the database.
- The different database management software packages handle the creation and use of relationships in different manners.

Cont..

- Depending on the type of interaction, the relationships are classified into three categories:
- **One-to-one relationship**
- **One-to-many relationship**
- **Many-to-many relationship**

One-to-one relationship

- A one-to-one relationship is written as **1:1** in short form. It exists between two entity sets, X and Y, if an entity in entity set X has only one matching entity in entity set Y, and vice versa.
- For example, a department in a college has one chairperson, and a chairperson chairs one department in a college.
- An employee manages one department in a company, and only one employee manages a department.

One-to-many relationship

- A one-to-many relationship is written as **1:M**.
- It exists between two entity sets, X and Y, if an entity in entity set X has many matching entities in entity set Y but an entity in entity set Y has only one matching entity in entity set X. In such a situation, a 1:M relationship exists between entity sets X and Y.

CONT…

- For example, a faculty teaches for one division in a college, but a division has many faculty members. The relationship between DIVISION and FACULTY is 1:M.
- An employee works in a department, but a department has many employees. The relationship between DEPARTMENT and EMPLOYEE is 1:M.

Many-to-many relationship

- A many-to-many relationship is written as **M:N or M:M**. It exists between two entity sets, X and Y, if an entity in entity set X has many matching entities in entity set Y and an entity in entity set Y has many matching entities in entity set X.
- For example, a student takes many courses, and many students take a course. An employee works on many projects, and a project has many employees.

DATABASE MANAGEMENT SYSTEM (DBMS)

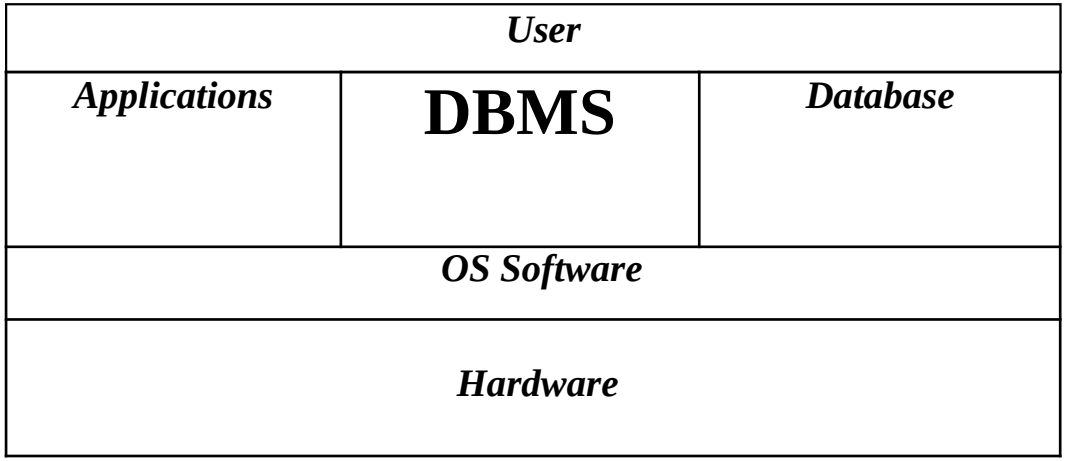
- The database system consists of the following components:
  - A database management System (DBMS) software package such as Microsoft Access, Visual Fox Pro, Microsoft SQL-Server, or Oracle.
  - A user-developed and implemented database or databases that include tables, a data dictionary, and other database objects.
  - Custom applications such as data-entry forms, reports, queries, blocks, and programs.

CONT..

- Computer hardware personal computers, mini computers,and mainframes in a network environment.
- Software—an operating system and a network operating system.
- Personnel a database administrator, a database designer/analyst, a programmer, and end users

Cont..

- Data are the raw materials. Information is processed, manipulated, collected, or organized data.



Cont..

- The information is produced when a user uses the applications to transform data managed by the DBMS.
- The database system is utilized as a decision-making system and is also referred to as an information system (IS).

Cont..

- A DBMS based on the relational model is also known as a Relational Database Management System (RDBMS).
- An RDBMS not only manages data but is also responsible for other important functions:
- It manages the data and relationships stored in the database. It creates a Data Dictionary as a user creates a database.
- The Data Dictionary is a system structure that stores Metadata (data about data).
- The Metadata include table names, attribute names, data types, physical space, relationships, and so on

Cont…

An RDBMS not only manages data but is also responsible for other important functions:

- It manages the data and relationships stored in the database.
- It creates a Data Dictionary as a user creates a database.
- The Data Dictionary is a system structure that stores Metadata (data about data).
- The Metadata include table names, attribute names, data types, physical space, relationships, and so on

Cont..

- It manages all day-to-day transactions
- It performs bookkeeping duties, so the user has data independence at the application level. The applications do not have information about data characteristics
- It transforms logical data requests to match physical data structures.
- When a user requests data, the RDBMS searches through the Data Dictionary, filters out unnecessary data, and displays the results in a readable and understandable form.

Cont..

- It allows users to specify validation rules. For example, if only M and F are possible values for the attribute gender, users can set validation rules to keep incorrect values from being accepted
- It secures access through passwords, encryption, and restricted user rights.
- It provides backup and recovery procedures for physical security of data.
- It allows users to share data with data-locking capabilities.

Cont..

- It provides import and export utilities to use data created in other database or spreadsheet software or to use data in other software.
- It enables users to join tables to view information stored in different tables within the database. The user is able to design a database with less redundancy, which means fewer data-entry errors, fewer data corrections, better data integrity, and a more efficient database

RELATIONAL DATABASE MODEL

- The need for data is always present.
- The computer age, the need to represent data in an easy-to-understand, logical form has led to many different models, such as the relational model, the hierarchical model, the network model, and the object model.
- Because of its simplicity in design and ease in retrieval of data, the relational database model has been very popular, especially in the personal computer environment

Cont..

- E. F. Codd developed the relational database model in 1970. The model is based on mathematical set theory, and it uses a **relation** as the building block of the database.
- The relation is represented by a two- dimensional, flat structure known as a table.
- The user does not have to know the mathematical details or the physical aspects of the data, but the user views the data in a logical, two-dimensional structure

Cont..

- The database system that manages a relational database environment is known as a Relational Database Management System (RDBMS).
- Some of the popular relational database systems are Oracle9i by Oracle Corporation, Microsoft Access 2000, and Microsoft Visual Fox Pro 6.0

Cont..

- A table is a matrix of rows and columns in which each row represents an entity and each column represents an attribute.
- In other words, a table represents an entity set as per database theory, and it represents a relation as per relational database theory.
- In daily practice, the terms table, relation, and entity set are used interchangeably

Cont..

PROJ2002

ProjNo	Loc	Customer
1	Miami	Stocks
3	Trenton	Smith
5	Phoenix	Robins
6	Edison	Shaw
7	Seattle	Douglas

PROJ2003

ProjNo	Loc	Customer
1	Miami	Stocks
2	Orlando	Allen
3	Trenton	Smith
4	Charlotte	Jones

PRJPARTS

ProjNo	PartNo	Qty
1	11	20
2	33	5
3	11	7
1	22	10
2	11	3

PARTS

PartNo	PartDesc	Vendor	Cost
11	Nut	Richards	19.95
22	Bolt	Black	5.00
33	Washer	Mobley	55.99

DEPARTMENT

DeptNo	DeptName
10	Production
20	Supplies
30	Marketing

EMPLOYEE

EmpNo	Ename	DeptNo	ProjNo	Salary
101	Carter	10	1	25000
102	Albert	20	3	37000
103	Breen	30	6	50500
104	Gould	20	5	23700
105	Barker	10	7	75000

Figure 1-2 Relational database tables.

INTEGRITY RULES

- In any database managed by an RDBMS, it is very important that the data in the underlying tables be consistent. If consistency is compromised. the data are not usable.
- This need led the pioneers of database field to formulate two integrity rules:
- **Entity integrity**
- **Referential integrity**

Entity integrity

- No column in a primary key may be null.
- The primary key provides the means of uniquely identifying a row or an entity.
- A null value means a value that is not known, not entered, not defined, or not applicable.
- A zero or a space is not considered to be a null value. If the primary key value is a null value in a row, we do not have enough information about the row to uniquely identify it.
- The RDBMS software strictly follows the entity integrity rule and does not allow users to enter a row without a unique value in the primary key column.

Referential integrity

- A foreign key value may be a null value, or it must exist as a value of a primary key in the referenced table.
- Referential integrity is not fully supported by all commercially available systems, but Oracle supports it religiously! Oracle does not allow you to declare a foreign key if it does not exist as a primary key in another table

Cont…

- It allows you to leave the foreign key column value as a null.
- If a user enters a value in the foreign key column, Oracle cross- references the referenced primary key column in the other table to confirm the existence of such a value.

Cont…

- It is not a good practice to use null values in any non-primary key columns, because this results in extra overhead on the system's part in search operations.
- The programmers or query users have to add extra measures to include or exclude rows with null values.
- In certain cases, it is not possible to avoid null values.

Cont…

- For example, an employee does not have a middle initial, an employee is hired but does not have an assigned department, or a student's major is undefined.
- In Oracle, a default value can be assigned to a column, and a user does not have to enter a value for that column.

THEORETICAL RELATIONAL LANGUAGES

E. F. Codd suggested two theoretical relational languages to use with the relational model:

- **Relational algebra,** a procedural language.
- **Relational calculus,** a nonprocedural language.

Cont…

- Third-generation high-level compiler languages can be used to manipulate data in a table, but they can only work with one row at a time.
- In contrast, the relational languages can work on the entire table or on a group of rows

Cont…

- The multiple-row manipulation does not even need a looping structure! The relational languages provide more power with a very little coding.

- Codd proposed these languages to embed them in other host languages for more processing capability and more sophisticated application development.

- In the database systems available today, nonprocedural Structured Query Language (SQL) is used as a data-manipulation sublanguage.

- The theoretical languages have provided the basis for SQL.

Relational Algebra

- Relational algebra is a procedural language, because the user accomplishes desired results by using a set of operations in a sequence.
- It uses set operations on tables to produce new resulting tables.
- These resulting tables are then used for subsequent sequential operations.
- In Oracle, all operation names are not actually used as programming terms, and most of these operations do not create a new resulting table, as shown in the following examples using relational algebra.

Relational Algebra

The nine operations used by relational algebra are:

1. Union.
2. Intersection.
3. Difference.
4. Projection.
5. Selection.
6. Product.
7. Assignment.
8. Join.
9. Division.

Union

- The union of two tables results in retrieval of all rows that are in one or both tables.
- The duplicate rows are eliminated from the resulting table.
- The resulting table does not contain two rows with identical data values. There is a basic requirement to perform a union operation on two tables

Cont..

- Both tables must have the same degree.
- The domains of the corresponding columns in two tables must be same.
- Such tables are said to be union compatible. In mathematical set theory, a union can be performed on any two sets, but in relational algebra, a union can be performed only on union-compatible tables.

Cont…

TABLE_A = PROJ2002 U PROJ2003

TABLE_A

ProjNo	Loc	Customer
1	Miami	Stocks
2	Orlando	Allen
3	Trenton	Smith
4	Charlotte	Jones
5	Phoenix	Robins
6	Edison	Shaw
7	Seattle	Douglas

Intersection

- The intersection of two tables produces a table with rows that are in both tables.
- The two tables must be union compatible to perform an intersection on them.

TABLE_B = PROJ2002 Ω PROJ2003

TABLE_B

ProjNo	Loc	Customer
1	Miami	Stocks
3	Trenton	Smith

Difference

- The difference of two tables produces a table with rows that are present in the first table but not in the second table. The difference can be performed on union-compatible tables only.

TABLE_C = PROJ2002 – PROJ2003

TABLE_C

ProjNo	Loc	Customer
5	Phoenix	Robins
6	Edison	Shaw
7	Seattle	Douglas

Cont…

TABLE_D = PROJ2003 – PROJ2002

the resulting TABLE_D will look like this:

TABLE_D

ProjNo	Loc	Customer
2	Orlando	Allen
4	Charlotte	Jones

Projection

- The projection operation allows us to create a table based on desirable columns from all existing columns in a table.
- The undesired columns are ignored. The projection operation returns the "vertical slices" of a table.
- The projection is indicated by including the table name and a list of desired columns

Cont…

TABLE_E = PARTS (PartDesc, Cost)

TABLE_E

PartDesc	Cost
Nut	19.95
Bolt	5.00
Washer	55.99

Selection

- The selection operation selects rows from a table based on a condition or conditions.
- The conditional operators (=, <>,>, >=, <, <=) and the logical operators (AND, OR, NOT) are used along with columns and values to create conditions.
- The selection operation returns "horizontal slices" from a table.

Cont…

TABLE_F = Sel (PARTS: Cost>10.00)

TABLE_F

PartNo	PartDesc	Vendor	Cost
11	Nut	Richards	19.95
33	Washer	Mobley	55.99

Product

- A product of two tables is a combination everything in both tables. It is also known as a Cartesian product.
- It can cause huge results with big tables. If the first table has x rows and the second table has y rows, the resulting product has x • y rows.
- If the first table has m columns and the second table has n columns, the resulting product has m + n columns.

Cont…

DEPARTMENT

DeptName
Production
Supplies
Marketing

EMPLOYEE

Ename
Carter
Albert

TABLE_G = EMPLOYEE • DEPARTMENT

TABLE_G

Ename	DeptName
Carter	Production
Carter	Supplies
Carter	Marketing
Albert	Production
Albert	Supplies
Albert	Marketing

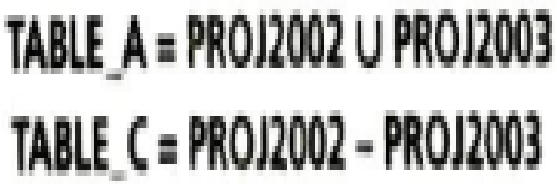
Assignment

- This operation creates a new table from existing tables.
- We have been doing it throughout all the other operations.
- Assignment (=) gives us an ability to name new tables that are based on other tables. Note that assignment is not an Oracle term.

Cont..

For example,

- TABLE_A = PROJ2002 U PROJ2003
  TABLE_C = PRO12002 - PROJ2003



Join

- The join is one of the most important operations because of its ability to get related data from a number of tables.

- The join is based on common set of values, which does not have to have the same name in both tables but does have to have the same domain in both tables. When a join is based on equality of value, it is known as a **natural join**.

- In Oracle, you will learn about the natural join, or **equijoin**.

- And also about other types of joins, such as **outer join, non equijoin**, and ,**self-join**, that are based on the operators other than the equality operator.

Cont..

TABLE_H = join (EMPLOYEE, DEPARTMENT : DeptNo = DeptNo)

TABLE_H

EmpNo	Ename	DeptNo	ProjNo	Salary	DeptName
101	Carter	10	1	25000	Production
102	Albert	20	3	37000	Supplies
103	Breen	30	6	50500	Marketing
104	Gould	20	5	23700	Supplies
105	Barker	10	7	75000	Production

Division

- The division operation is the most difficult operation to comprehend.
- It is not as simple as division in mathematics. In relational algebra, it identifies rows in one table that have a certain relationship to all rows in another table. Let us consider the following two tables.

Cont..

PROJ

ProjNo
1
2
3

PRJPARTS

ProjNo	PartNo
1	11
2	33
3	11
1	22
2	11

TABLE_I = PRJPARTS / PROJ

TABLE_I

PartNo
11

Database Design: Data Modeling and Normalization

- Data Modeling
- Dependency
- Database Design
- Normal forms
- Dependency Diagrams
- Renormalizations
- Another Example of Normalization

DATA MODELING

- A model is a simplified version of real-life, complex objects.
- Databases are complex, and data modeling is a tool to represent the various components and their relation-ships
- The entity-relationship (E-R) model is a very popular modeling tool among many such tools available today. Many tools are available for data modeling with E-R

Cont…

- All tools have some variations in representation of components.
- The E- R model provides:

✓ An excellent communication tool.

✓ A simple graphical representation of data.

Cont…

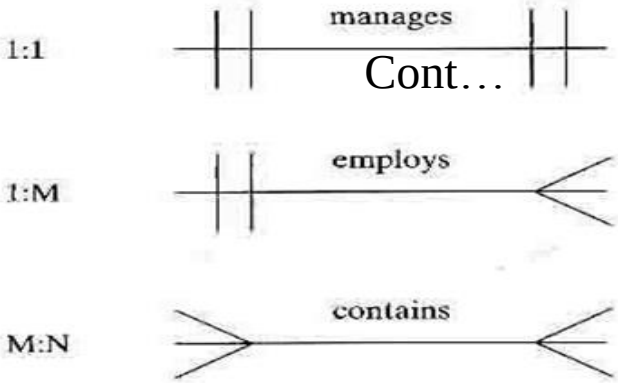
- The E-R model uses E-R diagrams (ERD) for graphical representation of the database components.
- An entity (or an entity set) is represented by a rectangle.
- The name of the entity (set) is written within the rectangle.
- Some tools prefer to use uppercase letters only for entities. The name of an entity set is a singular noun. For example, EMPLOYEE, CUSTOMER, and DEPARTMENT are singular entity set names.

Cont…

- A line represents relationship between the two entities.
- The name of the relationship is an active verb in lowercase letters.
- For example, works; manages, and employs are active verbs. Passive verbs can be used, but active verbs are preferable

Cont…

Entity representation in an E-R diagram.



Representation of relationship in an E-R diagram.

Cont…

Cont…

- The types of relationships (1:1, 1:M, and M:N) between entities are called connectivity or multiplicity.
- The connectivity is shown with vertical or angled lines next to each entity, For example, an EMPLOYEE supervises a DEPARTMENT, and a DEPARTMENT has one EMPLOYEE supervisor.
- A DIVISION contains many FACULTY members, but a FACULTY works for one DIVISION. An INVOICE contains many ITEMs, and an ITEM can be in more than one INVOICE.

Cont…

- Let us put everything together and represent these scenarios with the E-R dia- gram that shows entities, relationships, and connectivity
- The relationship between two entities can be given using the lower and upper limits. This information is called the **cardinality.**
- The cardinality is written next to each entity in the form *(n, m)*, where *n* is the minimum number and *m* is the maximum number

Cont…

- For example, (1,1) next to EMPLOYEE means that an employee can supervise a minimum of one and a maximum of one department.
- Similarly, (1,1) next to DEPARTMENT says that one and only one employee supervises the department.
- The value (1,N) means a minimum of one and a maximum equal to any number. Some modern tools do not show cardinality in an E-R diagram

Cont…

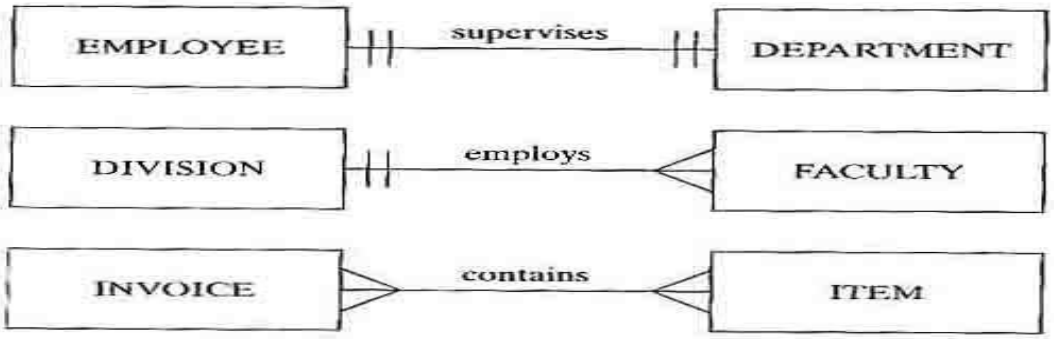
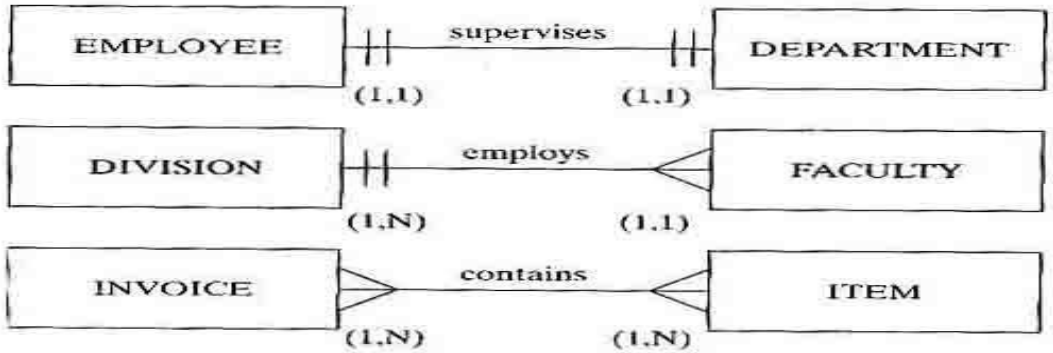


Figure 2-3 Entity, relationship, and connectivity.



Cardinality.

Cont…

- In reality, corporations set rules for the minimum and maximum values for cardinality.
- A corporation may decide that a department must have a minimum of 10 employees and a maximum of 25 employees, which results in cardinality of (10,25).
- A college decides that a computer-science course section must have at minimum 5 students to recover the cost incurred and at maximum 35 students, because the computer lab contains only 35 terminals

Cont…

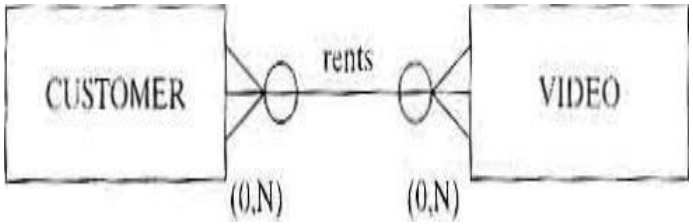
- An employee can be part of zero or more than one department, and an item may not be in any invoice! These types of decisions are known as business rules.
- The above E-R diagram with added cardinality. In real life, it is possible to have an entity that is not related to another entity at all times. The relationship becomes optional in such a case. In the example of a video rental store. a customer can rent video movies

Cont…

- In this case there are times when the customer has not rented any movie, and there are times when the customer has rented one or more movies.
- Similarly, there can be a movie in the database that is or is not rented at a particular time.

Cont…

- These are called optional relationships and are shown with a small circle next to the optional entity.
- The optional relationship can occur in 1:1, 1:M, or M:N relationships, and it can occur on one or both sides of the relationship



Cont…

- In relational databases, many-to-many (M:N) relationships are allowed, but they are not easy to implement.
- For example, an invoice has many items. and an item can be in many invoices. Refer to the INVOICE and ITEM relationship .
- At this point, you will he introduced to the relational schema, a graphical representation of tables, their column names, key components, and relations between the primary key in one table and the foreign key in another. You will also see the decomposition of an M:N relationship into two 1:M relationships

Cont…

- The decomposition from M:N to 1:M involves a third entity, known as a composite entity or an associative entity.
- The composite entity is created with the primary key from both tables with M:N relationships.
- The new entity has a composite key, which is a combination of primary keys from the original two entities. In the E-R diagram. a composite entity is drawn as a diamond within a rectangle.
- The composite entity has a composite primary key with two columns, each of them being foreign keys referencing the other two entities in the database.

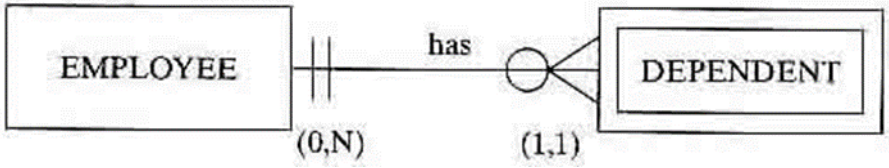
Cont..

- For example, the foreign key INVOICENO in the INVITEM table references the INVOICENO column in the INVOICE table, and the foreign key ITEMNO in the INVITEM table references the ITEMNO column in the ITEM table.
- In a database, there are entities that cannot exist by themselves.
- Such entities are known as weak entities. you will be introduced to two different sample databases.
- In the employee database of that chapter, there is an entity called EMPLOYEE with employees' demographic information and another entity called DEPENDENT with information about each employee's dependents.

Cont…

- The DEPENDENT entity cannot exist by itself. There are no dependents for an employee who does not exist.
- In other words, you need the existence of an employee for his or her dependent to exist in the database. The weak entities are shown by double lined rectangles

Cont…



Weak Entity

- Some of the other elements considered in the database design are:

  Simple attributes—attributes that cannot be subdivided; for example, last name, city, or gender.

  Composite attributes—attributes that can be subdivided, into atomic form; for example, a full name can be subdivided into the last name, first name, and middle initial.

Cont..

- Single-valued attributes—attributes with a single value; for example, Employee ID, Social Security number, or date of birth.
- Multivalued attributes—attributes with multiple values; for example, degree codes or course registration. The multivalued attributes have to be given special consideration.
- They can be entered into one attribute with a value separator mark, or they can be entered in separate attributes with names like Course l, Course2, Course3, and so on. Alternatively, a separate, composite entity can be created

DEPENDENCY

- Every table in the database should have a primary key, which uniquely identifies an entity.
- For example, PartNo is a primary key in the PARTS table, and DeptNo is a primary key in the DEPARTMENT table.
- In Oracle, if you create a table and do not define its primary key.
- Oracle does not consider it to be an error. You should define a primary key for all tables for integrity of data. Each table has other columns that do not make up the primary key for the table.
- Such columns are called the non key columns. The non key columns are functionally dependent on the primary key column.

Cont…

- For example, Part Desc and Cost in the PARTS table are dependent on the primary key Par tNo, and DeptName is dependent on the primary key DeptNo in the DEPARTMENT table
- Now, let us take a scenario as shown below. The INVOICE table in does not have any single column that can uniquely identify an entity.
- The first choice would be InvNo. It is not a unique value in the table, however, because an invoice may contain more than one item and there may be more than one entry for an invoice.
- CustNo cannot be the primary key, because there can be many invoices for a customer and CustNo does not identify an invoice. ItemNo cannot be the primary key either, because an item may appear in more than one invoice and ItemNo does not describe an invoice.
- The table has a composite primary key, which consists of InvNo and ItemNo. InvNo and ItemNo together make up unique values for each row. All other columns that do not constitute the primary key are nonkey columns, and they are dependent on the primary key.

INVOICE

Inv No	InvDate	CustN	ItemNo	CustName	ItemName	ItemPrice	Qty
1001	04/14/03	212	1	Starks	Screw	$2.25	5
1001	04/14/03	212	3	Starks	Bolt	S3.99	5
1001	04/14/03	212	5	Starks	Washer	$1.99	9
1002	04/17/03	225	1	Connors	Screw	$2.25	2
1002	04/17/03	225	2	Connors	Nut	$5.00	3
1003	04/17/03	239	1	Kapur	Screw	$2.25	7
1003	04/17/03	239	2	Kapur	Nut	$5.00	1
1004	04/18/03	211	4	Garcia	Hammer	$9.99	5

INVOICE table and its columns.

Cont..

There are three types of dependencies in a table:

- ***Total or full dependency:*** A nonkey column dependent on all primary key columns shows total dependency.
- ***Partial dependency:*** In partial dependency, a nonkey column is dependent on part of the primary key.
- ***Transitive dependency:*** In transitive dependency, a nonkey column is dependent on another nonkey column.

Cont…

- For example, in the INVOICE table, Item Name and Item Price are non key columns that are dependent only on a part of the primary key column ItemNo.

- They are not dependent on the InvNo column. Similarly, the non key

- column Inv Date is dependent only on InvN o. They are partially dependent on the primary key columns. The nonkey column CustName is not dependent on any primary key column but is dependent on another non key column, CustNo. It is said to have transitive dependency. The non key column Qty is dependent on both InvNo and ItemNo, so it is said to have full dependency.

DATABASE DESIGN

- Relational database design involves an attempt to synthesize the database structure to get the "first draft."
- The initial draft goes through an analysis phase to improve the structure.
- More formal techniques are available for the analysis and improvement of the structure. In the synthesis phase, entities and their relationships are identified.

Cont…

- The characteristics or the columns of all entities are also identified, and the designer defines the domains for each column.
- The candidate keys are picked, and primary keys are selected from them.
- The minimal set of columns is used as a primary key. If one column is sufficient to uniquely identify art entity, there is no need to select two columns to create a composite key

Cont…

- Avoid using names as primary keys, and break down composite columns into separate columns.
- For example, a name should be split into last name and first name.
- Once entities, columns, domains, and keys are defined, each entity is synthesized by creating a table for it.
- A process called normalization analyzes tables created by the synthesis process.

NORMAL FORMS

- Data are repeated from row to row.
- For example, InvDate, CustNo, and CustName are repeated for same InvNo.
- The ItemName is entered repeatedly from invoice to invoice.
- There is a large amount of redundant data in a table with just eight rows! Redundant data can pose a huge problem in databases.
- First of all, someone has to enter the same data repeatedly. Second, if a change is made in one piece of the data, the change has to be made in many places

NORMAL FORMS

- For example, if customer Starks changes his or her name to Starks-Johnson, you would go to the individual row in INVOICE and make that change. The redundancy may also lead to **anomalies**.

Anomalies

- A deletion anomaly results when the deletion of information about one entity leads to the deletion of information about another entity.
- For example, if an invoice for customer Garcia is removed, information about item number 4 is also deleted.
- An insertion anomaly occurs when the information about an entity cannot be inserted unless the information about another entity is known.
- For example, if the company buys a new item, this information cannot be entered unless an invoice is created for a customer with that new item. An update anomaly can occur if the item price changes to a new price. The price change is valid after the change date, but not before the change date.

Cont…

- Unnecessary and unwanted redundancy and anomalies are not appropriate in databases. Such tables are in lower normal form.
- Normalization is a technique to re-duce redundancy. It is a decomposition process to split tables.
- The splitting is per-formed carefully so that no information is lost. The higher the normal form is, the lower the redundancy. The table in is in first normal form (INF).

First Normal Form (INF)

- A table is said to be in first normal form, or can be labeled INF, if the following conditions exist:
- The primary key is defined. This includes a composite key if a single column cannot be used as a primary key. In our INVOICE table, InvNo and ItemId are defined as the composite primary key components.
- All nonkey columns show functional dependency on the primary key components. If you know the invoice number and the item number, you can find out the invoice date, customer number and name, item name and price, and quantity ordered. For example, if InvNo = 1001 and ItemNo = 5 are

•

known, then InvDate = 04114/03. ItemName = Washer, ItemPrice = $1.99, CustNo = 212. and CustName = Starks.

Cont…

- The table contains no multivalued columns. In a single-valued column, the intersection of a row and a column returns only one value.
- In a normalized table, the intersection of a row and a column is a single value.
- Some database packages, such as Uni data and Prime Information, allow multiple values in a column in a row, but Oracle does not.

The INVOICE table of in un normalized form. The ItemNo, ItemName, ItemPrice, and Qty columns are multivalued

Cont…

- A table that is in INF may have redundant data.
- A table in 1NF does not show data consistency and integrity in the long run.
- The normalization technique is used to control and reduce redundancy and to bring the table to a higher normal form.

Cont…

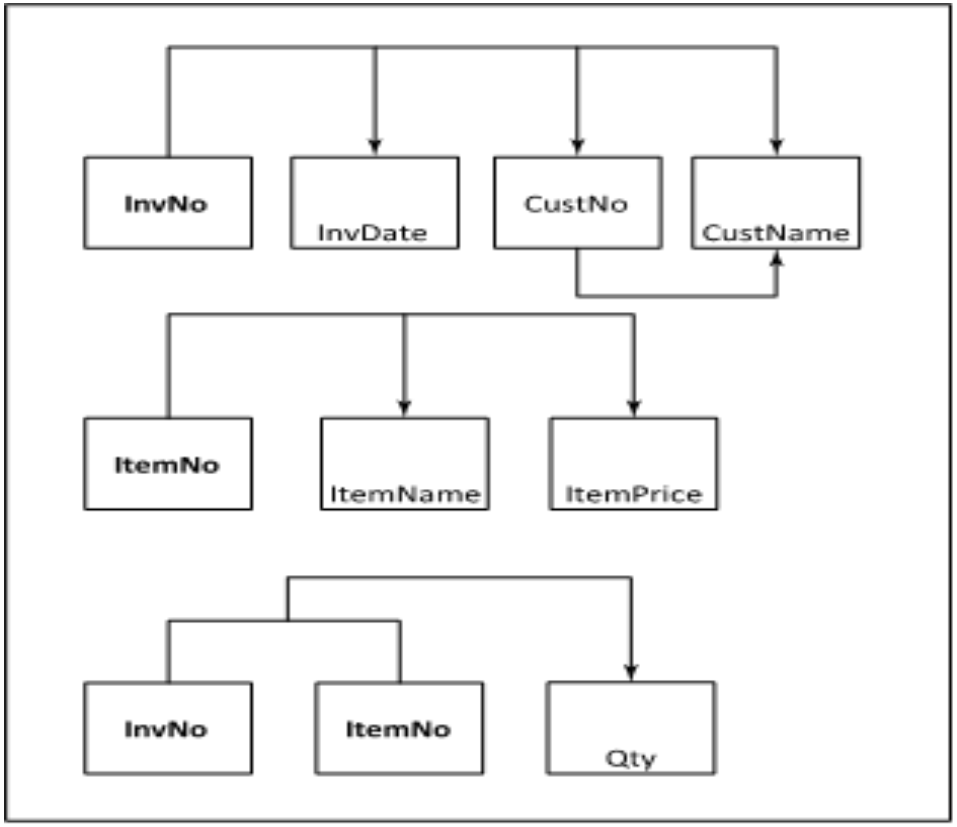
INVOICE

InvNo	InvDate	CustNo	ItemNo	CustName	ItemName	ItemPrice	Qty
1001	04/14/03	212	1	Starks	Screw	$2.25	5
			3		Bolt	$3.99	5
			5		Washer	$1.99	9
1002	04/17/03	225	1	Connors	Screw	$2.25	2
			2		Nut	$5.00	3
1003	04/17/03	239	1	Kapur	Screw	$2.25	7
			2		Nut	$5.00	1
1004	04/18/03	211	4	Garcia	Hammer	$9.99	5

Second Normal Form (2NF)

- A table is said to be in second normal form, or 2NF, if the following requirements are satisfied:
- All INF requirements are fulfilled.
- There is no partial dependency.

Cont…


InvNo
InvDate
CustNo
CustName
ItemNo
ItemName
ItemPrice
InvNo
ItemNo
Qty

Cont…

- As you already know, partial dependency exists in a table in which non key columns are partially dependent on part of a composite key.
- Suppose a table is in 1NF and does not have a composite key.
- Is it in the second normal form also? Yes, it is in 2NF, because there is no partial dependency. Partial dependency only exists in a table with a composite key

Third Normal Form (3NF)

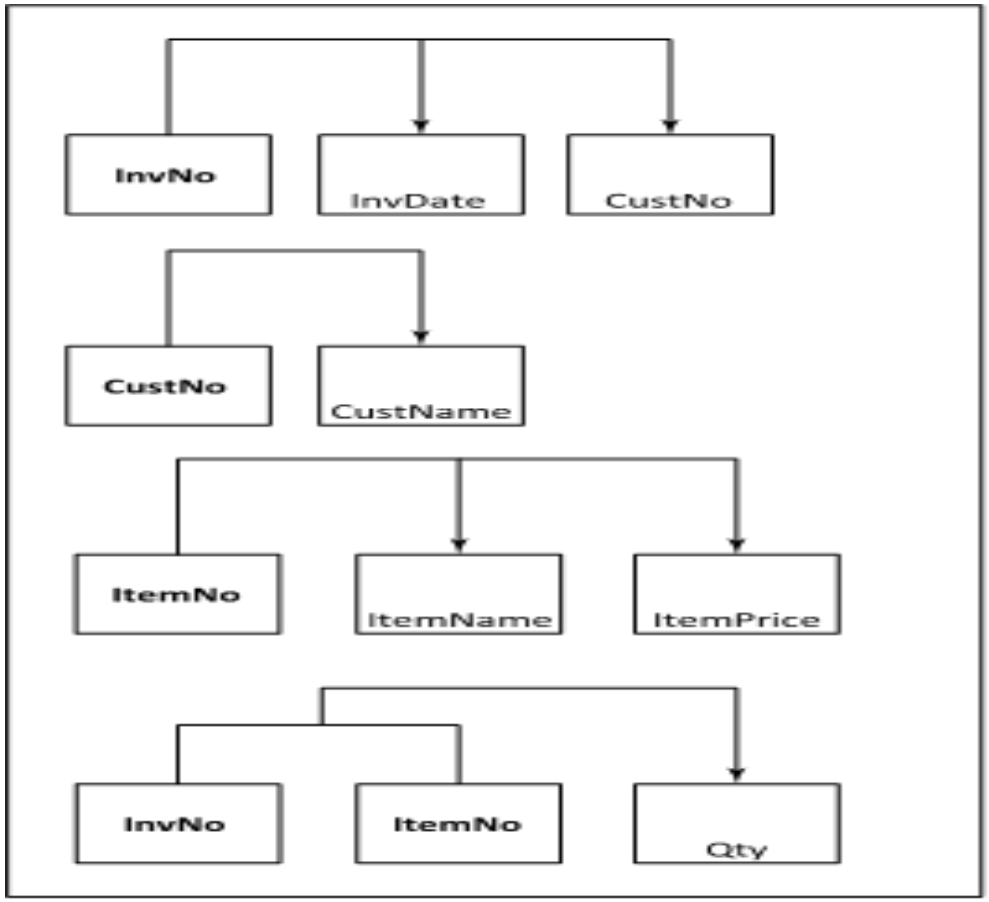
A table is said to be in third normal form, or 3NF, if the following requirements are satisfied:

- All 2NF requirements are fulfilled.
- There is no transitive dependency.

Cont…

- A table that has transitive dependency is not in 3NF, but it needs to be decomposed further to achieve 3NF.
- However, a table in 2NF that does not contain any transitive dependency does not need any further decomposition and is automatically in 3NF.
- Other, higher normal forms are defined in some database texts.
- Boyce-Codd normal form (BCNF), fourth normal form (4NF), fifth normal form (5NF), and do-main key normal form (DKNF) are not covered in this text. In the following section, you will learn the normalization process by using dependency diagrams.

Cont…

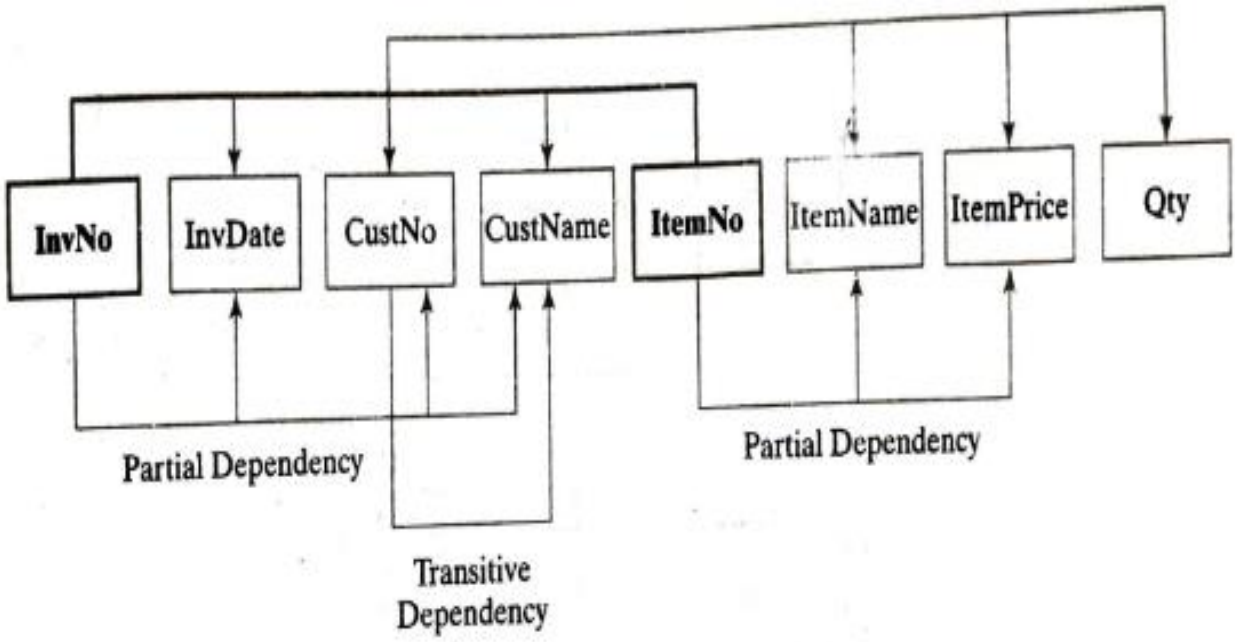

InvNo
InvDate
CustNo
CustNo
CustName
ItemNo
ItemName
ItemPrice
InvNo
ItemNo
Qty

Dependency Diagrams

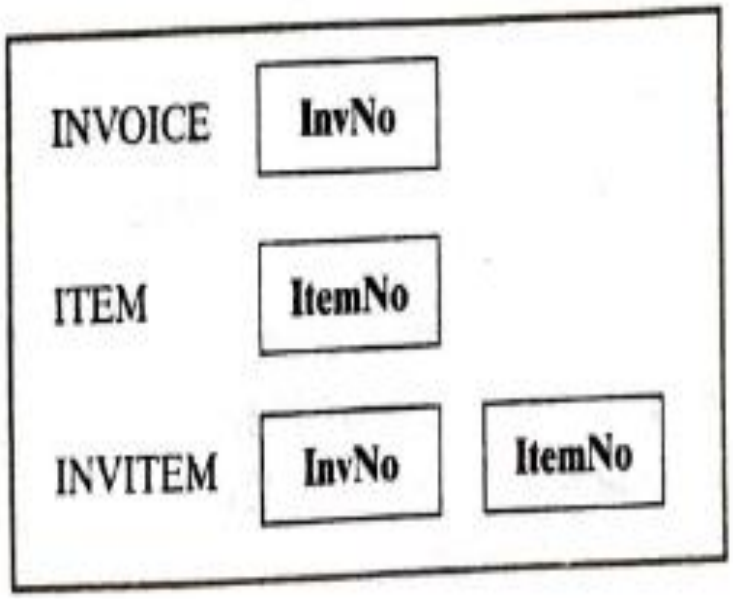
Dependency diagram is used to show total (Full), Partial and Transitive dependencies in a table)

- Conversion from 1NF to 2NF
- Conversion from 2NF to 3NF

Dependency diagram

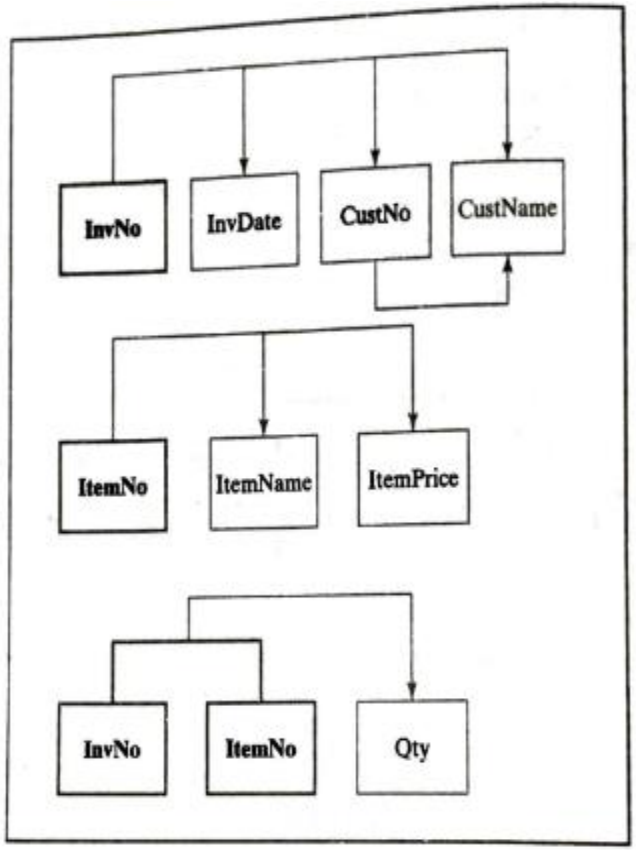


1NF to 2NF Decomposition

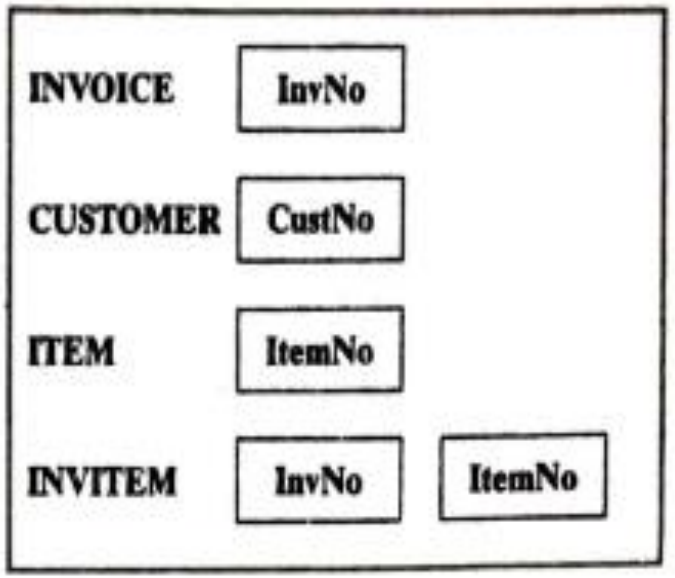


Conversion from 1NF to 2NF

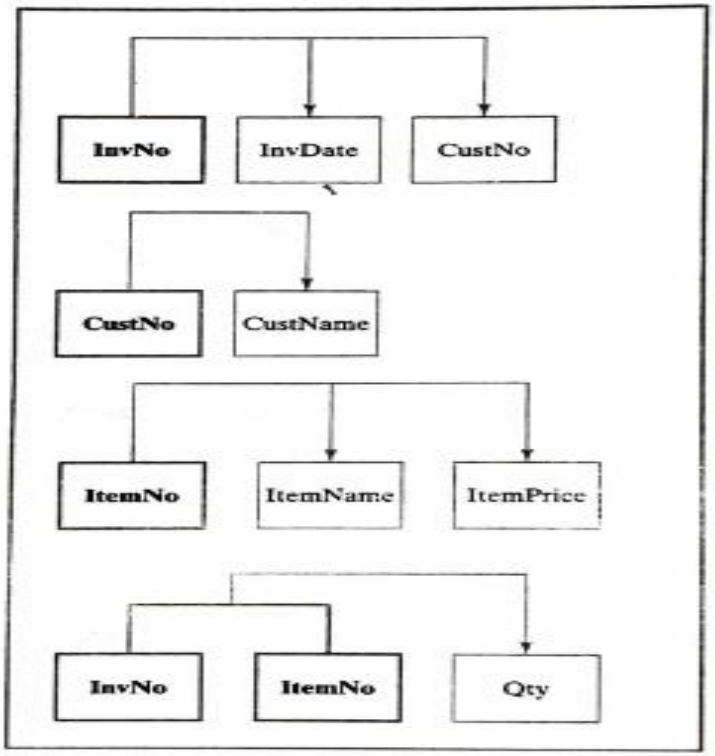
Tables in 2NF



Conversion from 2NF to 3NF



Tables in 3NF



Denormalization

- The normalization process splits tables into smaller tables.
- These tables are joined through common columns to retrieve information from different tables.
- Denormalization is the reverse process.

Another Example of Normalization

Playerid	Playername	Year	JerseyNum	BirthDate	PointsScoredinYear	GamesPlayed	Teamid	TeamName	TeamLoc
1	JOHNSON	2001	32	4/15/1980	150	5	1	MUSTANGS	BRONX
1	JOHNSON	2002	32	4/15/1980	174	6	1	MUSTANGS	BRONX
1	JOHNSON	2003	32	4/15/1980	115	5	1	MUSTANGS	BRONX
2	NAMAN	2001	10	12/2/1985	100	3	1	MUSTANGS	BRONX
2	NAMAN	2002	10	12/2/1985	149	6	2	DEVILS	PRINCETON
2	NAMAN	2003	10	12/2/1985	185	6	5	EAGLES	BRUNSWICK
3	SHAW	2001	11	5/10/1986	99	5	4	BEARCATS	FORDS
3	SHAW	2002	11	5/10/1986	97	6	4	BEARCATS	FORDS
3	SHAW	2003	3	5/10/1986	115	6	6	KINGS	MANHATTAN
4	ALBERT	2003	33	5/19/1983	29	3	3	BULLDOGS	MONROE
5	ANTHONY	2001	21	1/19/1979	110	6	3	BULLDOGS	MONROE
5	ANTHONY	2002	21	1/19/1979	78	4	3	BULLDOGS	MONROE
5	ANTHONY	2003	33	1/19/1979	111	5	1	MUSTANGS	BRONX
6	RICHARDS	2003	33	7/10/1977	63	6	2	DEVILS	PRINCETON
7	ROBERTS	2003	55	6/6/1981	44	6	5	EAGLES	BRUNSWICK
8	JONES	2001	2	12/31/1981	123	6	6	KINGS	MANHATTAN
8	JONES	2002	2	12/31/1981	100	6	4	BEARCATS	FORDS
9	JORDAN	2003	23	2/17/1986	101	2	1	MUSTANGS	BRONX

- The table contains a composite key that is composed of two columns, PlayerId and Year. This table contains each player's yearly statistics as well as team information.
- A player may belong to different teams during different years (it is assumed that a player belongs to one team during a year).