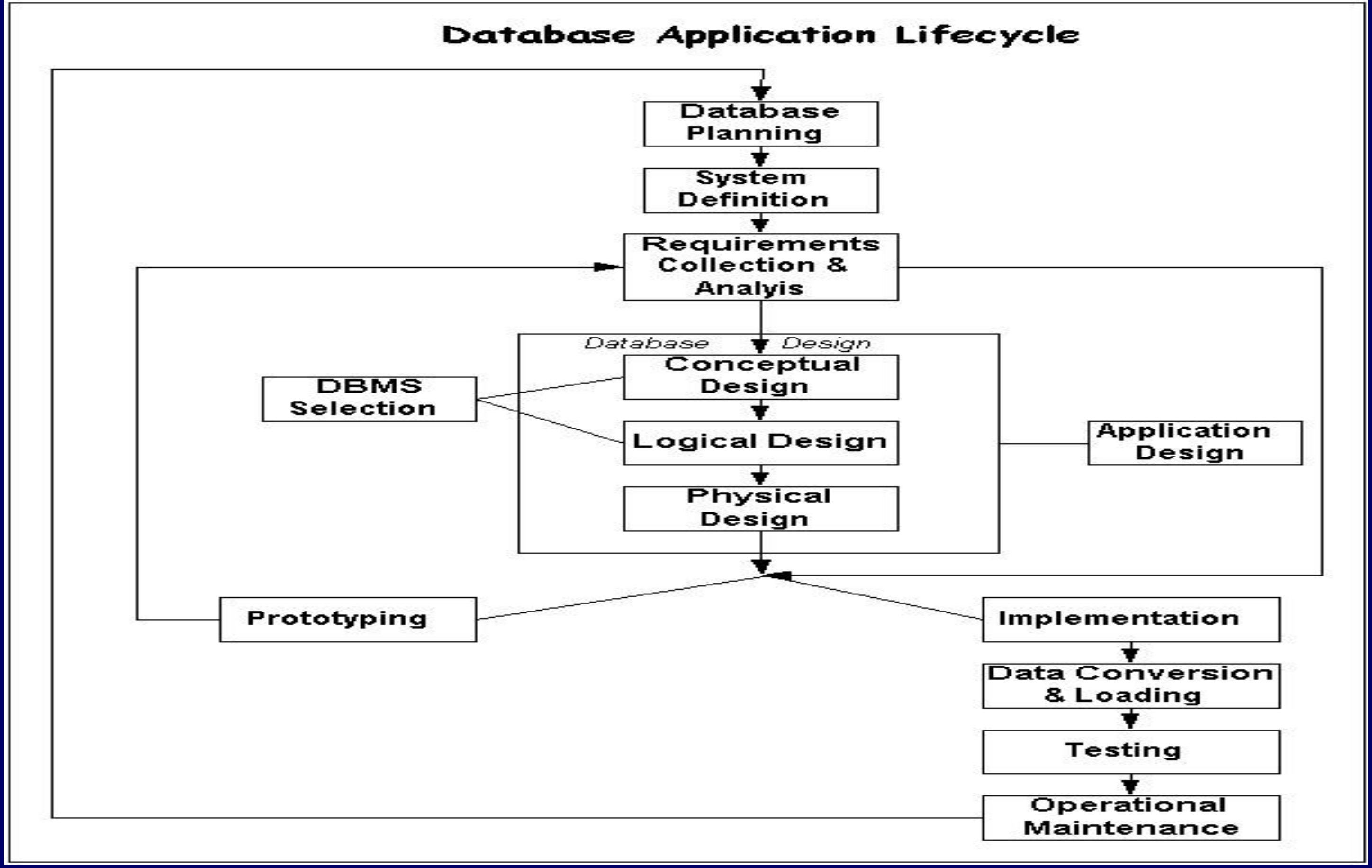
Unit III

Database Analysis and Design Techniques

Objective:

- □ **Phases in Database Development Life Cycle.**
- □ **Relational Database Design:**
- □ Purpose of Normalization
- □ The Process of Normalization

Database Development Life Cycle



Database Application Lifecycle

The database development process comprises a series of phases. The major phases in information engineering are:

- Planning
- System definition
- Requirement Analysis
- Application Design
- DBMS Selection
- Implementation
- Maintenance

Database Planning

- Focus here is on management activities
- Activities include:
  - Work to be done
  - Resources available

Database planning

- The database-planning phase begins when a customer requests to develop a database project.
- It is set of tasks or activities, which decide the resources required in the database development and time limits of different activities.
- During planning phase, four major activities are performed.
- Review and approve the database project request.
- Prioritize the database project request.
- Allocate resources such as money, people and tools.
- Arrange a development team to develop the database project.
- Database planning should also include the development of standards that govern how data will be collected, how the format should be specified, what necessary documentation will be needed.

System Definition

- Scope
- Parameters
- Application areas
- User groups

Requirements Analysis

- Requirements analysis is done in order to understand the problem, which is to be solved.
- The person responsible for the requirements analysis is often called "Analyst".
- In requirements analysis phase, the requirements and expectations of the users are collected and analyzed.
- There are two major activities in requirements analysis.
- Problem understanding or analysis
- Requirement specifications.

Goals of Requirements Analysis

- **to determine the data requirements of the database in terms of primitive objects**
- **to classify and describe the information about these objects**
- **to identify and classify the relationships among the objects**
- **to determine the types of transactions that will be executed on the database and the interactions between the data and the transactions**
- to identify rules governing the integrity of the data

Database design

- The process of creating a design for a database that will support the enterprise’s operations and objectives

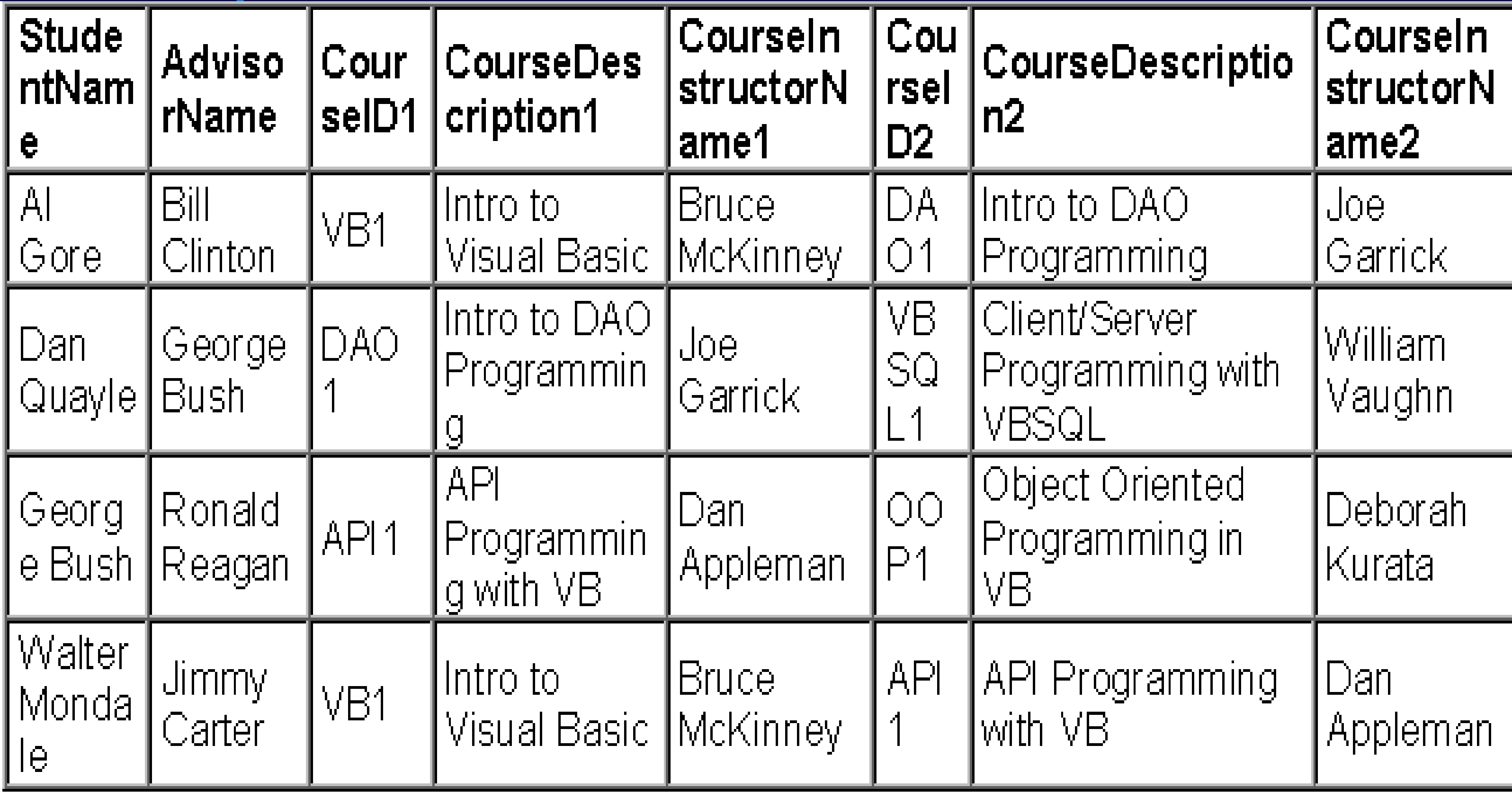
Database Design

- In this phase, the information models that were developed during analysis are used to design a conceptual schema for the database and to design transaction and application.
- In conceptual schema design, the data requirements collected in Requirement Analysis phase are examined and a conceptual database schema is produced.
- In transaction and application design, the database applications analyzed in Requirement Analysis phase are examined and specifications of these applications are produced.

Database Design Framework

- Determine the information requirements
- Analyse the real-world objects that you want to model in the database
- Determine primary key attributes
- Develop a set of rules that govern how each table is accessed, populated and updated
- Identify relationship between the entities
- Plan database security

Students and courses



StudentName	AdvisorName	CourseID1	CourseDescription1	CourseInstructorName1	CourseID2	CourseDescription2	CourseInstructorName2
Al Gore	Bill Clinton	VB1	Intro to Visual Basic	Bruce McKinney	DAO1	Intro to DAO Programming	Joe Garrick
Dan Quayle	George Bush	DAO1	Intro to DAO Programming	Joe Garrick	VBSQL1	Client/Server Programming with VBSQL	William Vaughn
George Bush	Ronald Reagan	API1	API Programming with VB	Dan Appleman	OOP1	Object Oriented Programming in VB	Deborah Kurata
Walter Mondale	Jimmy Carter	VB1	Intro to Visual Basic	Bruce McKinney	API1	API Programming with VB	Dan Appleman

Identifying entities

- Students
- Courses
- Instructors
- StudentCourses
- Advisors

Identifying attributes

□ What information on each entity should we know?

- •Students(student_id, Fname, lname, phone, advisor_id)
- •Advisors(Advisor_id, Advisorname, Advisorphone)
- •Instructors(instructor_id, Instructorname, Instructorphone)
- •StudentCourses(Student_id, Course_id)
- •Courses(Course_id, Coursedescription instructor_id)

DBMS Selection

- In this phase an appropriate DBMS is selected to support the information system.
- A number of factors are involved in DBMS selection. They may be technical and economical factors.
- The technical factors are concerned with the suitability of the DBMS for information system.
- The following technical factors are considered.
- Type of DBMS such as relational, object-oriented etc
- Storage structure and access methods that the DBMS supports.
- User and programmer interfaces available.
- Type of query languages.
- Development tools etc.

Implementation

- After the design phase and selecting a suitable DBMS, the database system is implemented.
- The purpose of this phase is to construct and install the information system according to the plan and design as described in previous phases.
- Implementation involves a series of steps leading to operational information system that includes creating database definitions (such as tables, indexes etc), developing applications, testing the system, developing operational procedures and documentation, training the users and populating the database.

Operational Maintenance

- Once the database system is implemented, the operational maintenance phase of the database system begins.
- The operational maintenance is the process of monitoring and maintaining the database system.
- Maintenance includes activities such as adding new fields, changing the size of existing field, adding new tables, and so on.
- As the database system requirement change, it becomes necessary to add new tables or remove existing tables and to reorganize some files by changing primary access methods or by dropping old indexes and constructing new ones.
- Some queries or transactions may be rewritten for better performance.
- Database tuning or reorganization continues throughout the life of database and while the requirements keep changing.

Relational Database Design

- The goal of relational database design is to generate a set of schemas that allow us to Store information without unnecessary redundancy.
- Retrieve information easily (and accurately).

Database Normalization

- **Normalization** is a technique for producing a set of relations with desirable properties, given the data requirements of an enterprise.
- The process of normalization is a formal method that identifies relations based on their primary or candidate keys and the functional dependencies among their attributes.

Purpose of Normalization

- Main purposes of normalization in a relational database:
- Eliminate redundancy
- Organize data efficiently
- Reduce the potential for data anomalies

Benefits of Normalization

- Less storage space
- Quicker updates
- Less data inconsistency
- Clearer data relationships
- Easier to add data
- Flexible Structure

Update Anomalies

- Relations that have redundant data may have problems called **update anomalies**, which are classified as ,
- Insertion anomalies
- Deletion anomalies
- Modification anomalies

Example of Update Anomalies

To insert a new staff with branchNo B007 into the StaffBranch relation;
To delete a tuple that represents the last member of staff located at a branch B007;
To change the address of branch B003.

StaffBranch

staffNo	sName	position	salary	branchNo	bAddress
SL21	John White	Manager	30000	B005	22 Deer Rd, London
SG37	Ann Beech	Assistant	12000	B003	163 Main St,Glasgow
SG14	David Ford	Supervisor	18000	B003	163 Main St,Glasgow
SA9	Mary Howe	Assistant	9000	B007	16 Argyll St, Aberdeen
SG5	Susan Brand	Manager	24000	B003	163 Main St,Glasgow
SL41	Julie Lee	Assistant	9000	B005	22 Deer Rd, London

Figure 1 StraffBranch relation

Example of Update Anomalies

Staff

staffNo	sName	position	salary	branceNo
SL21	John White	Manager	30000	B005
SG37	Ann Beech	Assistant	12000	B003
SG14	David Ford	Supervisor	18000	B003
SA9	Mary Howe	Assistant	9000	B007
SG5	Susan Brand	Manager	24000	B003
SL41	Julie Lee	Assistant	9000	B005

Branch

branceNo	bAddress
B005	22 Deer Rd, London
B007	16 Argyll St, Aberdeen
B003	163 Main St,Glasgow

Figure 2 Straff and Branch relations

Functional Dependencies

Identifying the primary key

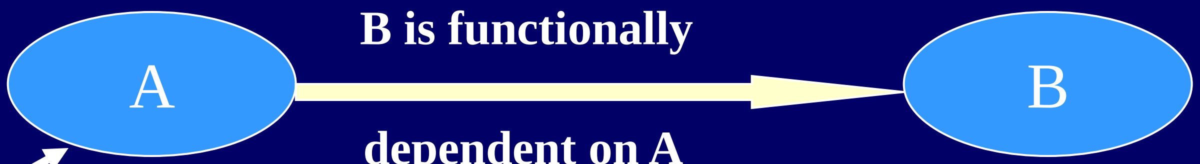
Functional dependency is a property of the meaning or semantics of the attributes in a relation. When a functional dependency is present, the dependency is specified as a **constraint** between the attributes.

An important integrity constraint to consider first is **the identification of candidate keys, one of which is selected to be the primary key** for the relation using functional dependency.

Functional Dependencies

Functional dependency describes the relationship between attributes in a relation.

For example, if A and B are attributes of relation R, and B is functionally dependent on A (denoted A → B), if each value of A is associated with exactly one value of B. (A and B may each consist of one or more attributes.)



Determinant

Refers to the attribute or group of attributes on the left-hand side of the arrow of a functional dependency

The Process of Normalization

- Normalization is often executed as a series of steps. Each step corresponds to a specific normal form that has known properties.

- As normalization proceeds, the relations become progressively more restricted in format, and also less vulnerable to update anomalies.

- For the relational data model, it is important to recognize that it is only first normal form (1NF) that is critical in creating relations. All the subsequent normal forms are optional.

First Normal Form (1NF)

Repeating group = (propertyNo, pAddress, rentStart, rentFinish, rent, ownerNo, oName)

Unnorm

A table that contains one or more r g groups.

ClientNo	cName	propertyNo	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	John kay	PG4 PG16	6 lawrence St,Glasgow 5 Novar Dr, Glasgow	1-Jul-00 1-Sep-02	31-Aug-01 1-Sep-02	350 450	CO40 CO93	Tina Murphy Tony Shaw
CR56	Aline Stewart	PG4 PG36 PG16	6 lawrence St,Glasgow 2 Manor Rd, Glasgow 5 Novar Dr, Glasgow	1-Sep-99 10-Oct-00 1-Nov-02	10-Jun-00 1-Dec-01 1-Aug-03	350 370 450	CO40 CO93 CO93	Tina Murphy Tony Shaw Tony Shaw

Figure 3 ClientRental unnormalized table

Definition of 1NF

First Normal Form is a relation in which the intersection of each row and column contains one and only one value.

There are two approaches to removing repeating groups from unnormalized tables:
1. Removes the repeating groups by entering appropriate data in the empty columns of rows containing the repeating data.

2. Removes the repeating group by placing the repeating data, along with a copy of the original key attribute(s), in a separate relation. A primary key is identified for the new relation.

1NF ClientRental relation with the first approach

The ClientRental relation is defined as follows,

ClientRental (clientNo, propertyNo, cName, pAddress, rentStart, rentFinish, rent, ownerNo, oName)

With the first approach, we remove the repeating group (property rented details) by entering the appropriate client data into each row.

ClientNo	propertyNo	cName	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	PG4	John Kay	6 lawrence St,Glasgow	1-Jul-00	31-Aug-01	350	CO40	Tina Murphy
CR76	PG16	John Kay	5 Novar Dr, Glasgow	1-Sep-02	1-Sep-02	450	CO93	Tony Shaw
CR56	PG4	Aline Stewart	6 lawrence St,Glasgow	1-Sep-99	10-Jun-00	350	CO40	Tina Murphy
CR56	PG36	Aline Stewart	2 Manor Rd, Glasgow	10-Oct-00	1-Dec-01	370	CO93	Tony Shaw
CR56	PG16	Aline Stewart	5 Novar Dr, Glasgow	1-Nov-02	1-Aug-03	450	CO93	Tony Shaw

Figure 4 1NF ClientRental relation with the first approach

1NF ClientRental relation with the second approach

With the second approach, we remove the repeating group (property rented details) by placing the repeating data along with a copy of the original key attribute (clientNo) in a separte relation.

Client (clientNo, cName)

PropertyRentalOwner (clientNo, propertyNo, pAddress, rentStart, rentFinish, rent, ownerNo, oName)

ClientNo	cName
CR76	John Kay
CR56	Aline Stewart

ClientNo	propertyNo	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	PG4	6 lawrence St,Glasgow	1-Jul-00	31-Aug-01	350	CO40	Tina Murphy
CR76	PG16	5 Novar Dr, Glasgow	1-Sep-02	1-Sep-02	450	CO93	Tony Shaw
CR56	PG4	6 lawrence St,Glasgow	1-Sep-99	10-Jun-00	350	CO40	Tina Murphy
CR56	PG36	2 Manor Rd, Glasgow	10-Oct-00	1-Dec-01	370	CO93	Tony Shaw
CR56	PG16	5 Novar Dr, Glasgow	1-Nov-02	1-Aug-03	450	CO93	Tony Shaw

Figure 5 1NF ClientRental relation with the second approach

Full functional dependency

Full functional dependency indicates that if A and B are attributes of a relation, B is fully functionally dependent on A if B is functionally dependent on A, but not on any proper subset of A.

A functional dependency A→B is **partially dependent** if there is some attributes that can be removed from A and the dependency still holds.

Second Normal Form (2NF)

Second normal form (2NF) is a relation that is in first normal form and every non-primary-key attribute is fully functionally dependent on the primary key.

The normalization of 1NF relations to 2NF involves the removal of partial dependencies. If a partial dependency exists, we remove the function dependent attributes from the relation by placing them in a new relation along with a copy of their determinant.

2NF ClientRental relation

The ClientRental relation has the following functional dependencies:

fd1 clientNo, propertyNo → rentStart, rentFinish (Primary Key)

fd2 clientNo → cName (Partial dependency)

fd3 propertyNo → pAddress, rent, ownerNo, oName (Partial dependency)

2NF ClientRental relation

After removing the partial dependencies, the creation of the three new relations called Client, Rental, and PropertyOwner.

Client (clientNo, cName)

Rental (clientNo, propertyNo, rentStart, rentFinish)

PropertyOwner (propertyNo, pAddress, rent, ownerNo, oName)

Client

ClientNo	cName
CR76	John Kay
CR56	Aline Stewart

Rental

ClientNo	propertyNo	rentStart	rentFinish
CR76	PG4	1-Jul-00	31-Aug-01
CR76	PG16	1-Sep-02	1-Sep-02
CR56	PG4	1-Sep-99	10-Jun-00
CR56	PG36	10-Oct-00	1-Dec-01
CR56	PG16	1-Nov-02	1-Aug-03

PropertyOwner

propertyNo	pAddress	rent	ownerNo	oName
PG4	6 lawrence St,Glasgow	350	CO40	Tina Murphy
PG16	5 Novar Dr, Glasgow	450	CO93	Tony Shaw
PG36	2 Manor Rd, Glasgow	370	CO93	Tony Shaw

Figure 6 2NF ClientRental relation

Third Normal Form (3NF)

Transitive dependency
A condition where A, B, and C are attributes of a relation such that if A → B and B → C, then C is transitively dependent on A via B (provided that A is not functionally dependent on B or C).

Third normal form (3NF)
A relation that is in first and second normal form, and in which no non-primary-key attribute is **transitively** dependent on the primary key.
The normalization of 2NF relations to 3NF involves the removal of transitive dependencies by placing the attribute(s) in a new relation along with a copy of the determinant.

3NF ClientRental relation

The functional dependencies for the Client, Rental and PropertyOwner relations are as follows:

Client

fd2	clientNo → cName	(Primary Key)

Rental

fd1	clientNo, propertyNo → rentStart, rentFinish	(Primary Key)
fd5	clientNo, rentStart → propertyNo, rentFinish	(Candidate key)
fd6	propertyNo, rentStart → clientNo, rentFinish	(Candidate key)

PropertyOwner

fd3	propertyNo → pAddress, rent, ownerNo, oName	(Primary Key)
fd4	ownerNo → oName	(Transitive Dependency)

3NF ClientRental relation

The resulting 3NF relations have the forms:

Client	(clientNo, cName)
Rental	(clientNo, propertyNo, rentStart, rentFinish)
PropertyOwner	(propertyNo, pAddress, rent, ownerNo)
Owner	(ownerNo, oName)

3NF ClientRental relation

Client

ClientNo	cName
CR76	John Kay
CR56	Aline Stewart

Rental

ClientNo	propertyNo	rentStart	rentFinish
CR76	PG4	1-Jul-00	31-Aug-01
CR76	PG16	1-Sep-02	1-Sep-02
CR56	PG4	1-Sep-99	10-Jun-00
CR56	PG36	10-Oct-00	1-Dec-01
CR56	PG16	1-Nov-02	1-Aug-03

PropertyOwner

propertyNo	pAddress	rent	ownerNo
PG4	6 lawrence St,Glasgow	350	CO40
PG16	5 Novar Dr, Glasgow	450	CO93
PG36	2 Manor Rd, Glasgow	370	CO93

Owner

ownerNo	oName
CO40	Tina Murphy
CO93	Tony Shaw

Figure 7 2NF ClientRental relation

Boyce-Codd Normal Form (BCNF)

- BCNF is a higher version of the Third Normal form
- This form deals with certain type of anomaly that is not handled by 3NF
- **Boyce-Codd normal form (BCNF)**
- **A relation is in BCNF, if and only if, every determinant is a candidate key.**
- *BCNF acts differently from 3NF only when there are multiple overlapping candidate keys.*

3 NF vs BCNF

- **The difference between 3NF and BCNF is that for a functional dependency A → B, 3NF allows this dependency in a relation if B is a primary-key attribute and A is not a candidate key, whereas BCNF insists that for this dependency to remain in a relation, A must be a candidate key.**
- BCNF is stricter than 3NF
- left side of any FD in the table must be a candidate key
- 3NF does not deal satisfactorily with the case of a relation with overlapping candidate keys i.e. composite candidate keys with at least one attribute in common

- Table must have more than one candidate keys
- Candidate keys must be composite
- Candidate keys must be disjoint (overlapping)