UNIVERSITY INSTITUTE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE &ENGINNERING

Bachelor of Engineering (Computer Science & Engineering)

Database Management System and CST-227

Prepared By: Ms. Heena Arora

Database Management System

Course Outcome

CO Number	Title	Level
CO1	Learn the fundamentals of database systems design and draw ER diagram for the real world problems.	Understand
CO2	Design and query database using SQL.	Apply
CO3	Analyze and apply concepts of normalization to relational database design	Understand
CO4	Learn the concept of transaction, concurrency and recovery.	Remember

OVERVIEW OF DATABASE

Introduction of Common Terms

Purpose of Database System

Data Dependence

Relational Database

Types of keys

Different types of Integrity

Common Terms

- **Data**: Facts, figures, statistics etc. having no particular meaning (e.g. 1, ABC, 19 etc).

- **Record**: Collection of related data items, e.g. in the above example the three data items had no meaning. But if we organize them in the following way, then they collectively represent meaningful information.

Table or Relation

- **Table or Relation**: Collection of related records,

  The columns of this relation are called Fields, Attributes or Domains. The rows are called Tuples or Records.

- **Database:** Collection of related relations. Consider the following collection of tables:

Files and Databases

- File: A collection of records or documents dealing with one organization, person, area or subject (Rowley)
  - Manual (paper) files
  - Computer files
- Database: A collection of similar records with relationships between the records (Rowley)
  - Bibliographic, statistical, business data, images, etc.

Purpose of Database Systems

- In the early days, database applications were built directly on top of file systems
- Drawbacks of using file systems to store data:
  - Data redundancy and inconsistency
    - Multiple file formats, duplication of information in different files
  - Difficulty in accessing data
    - Need to write a new program to carry out each new task
  - Data isolation — multiple files and formats
  - Integrity problems
    - Integrity constraints (e.g. account balance > 0) become "buried" in program code rather than being stated explicitly
    - Hard to add new constraints or change existing ones

Continued

- Drawbacks of using file systems (cont.)
  - Atomicity of updates
    - Failures may leave database in an inconsistent state with partial updates carried out
    - Example: Transfer of funds from one account to another should either complete or not happen at all
  - Concurrent access by multiple users
    - Concurrent accessed needed for performance
    - Uncontrolled concurrent accesses can lead to inconsistencies
      - Example: Two people reading a balance and updating it at the same time
  - Security problems
    - Hard to provide user access to some, but not all, data
- Database systems offer solutions to all the above problems

Importance of DBMS

- It helps make data management more efficient and effective.
- Its query language allows quick answers to ad hoc queries.
- It provides end users better access to more and better-managed data.
- It promotes an integrated view of organization’s operations -- “big picture.”
- It reduces the probability of inconsistent data.
- Data Independence
- Efficient data access
- Data integrity and security
- Data administration
- Concurrent access and crash recovery

Examples of Database Applications

- Banking: all transactions
- Airlines: reservations, schedules
- Universities: registration, grades
- Sales: customers, products, purchases
- Online retailers: order tracking, customized recommendations
- Manufacturing: production, inventory, orders, supply chain
- Human resources: employee records, salaries, tax deductions

Three Views of Data

- **Physical level:** describes how a record (e.g., customer) is stored.
- **Logical level:** describes data stored in database, and the relationships among the data.
- **View level:** application programs hide details of data types. Views can also hide information (such as an employee's salary) for security purposes.
- The database can be viewed from different levels of abstraction to reveal different levels of details. From a bottom-up manner, we may find that there are three levels of abstraction or views in the database.
- The term Abstraction is very important here. Generally it means the amount of detail you want to hide. Any entity can be seen from different perspectives and levels of complexity to make it a reveal its current amount of abstraction.

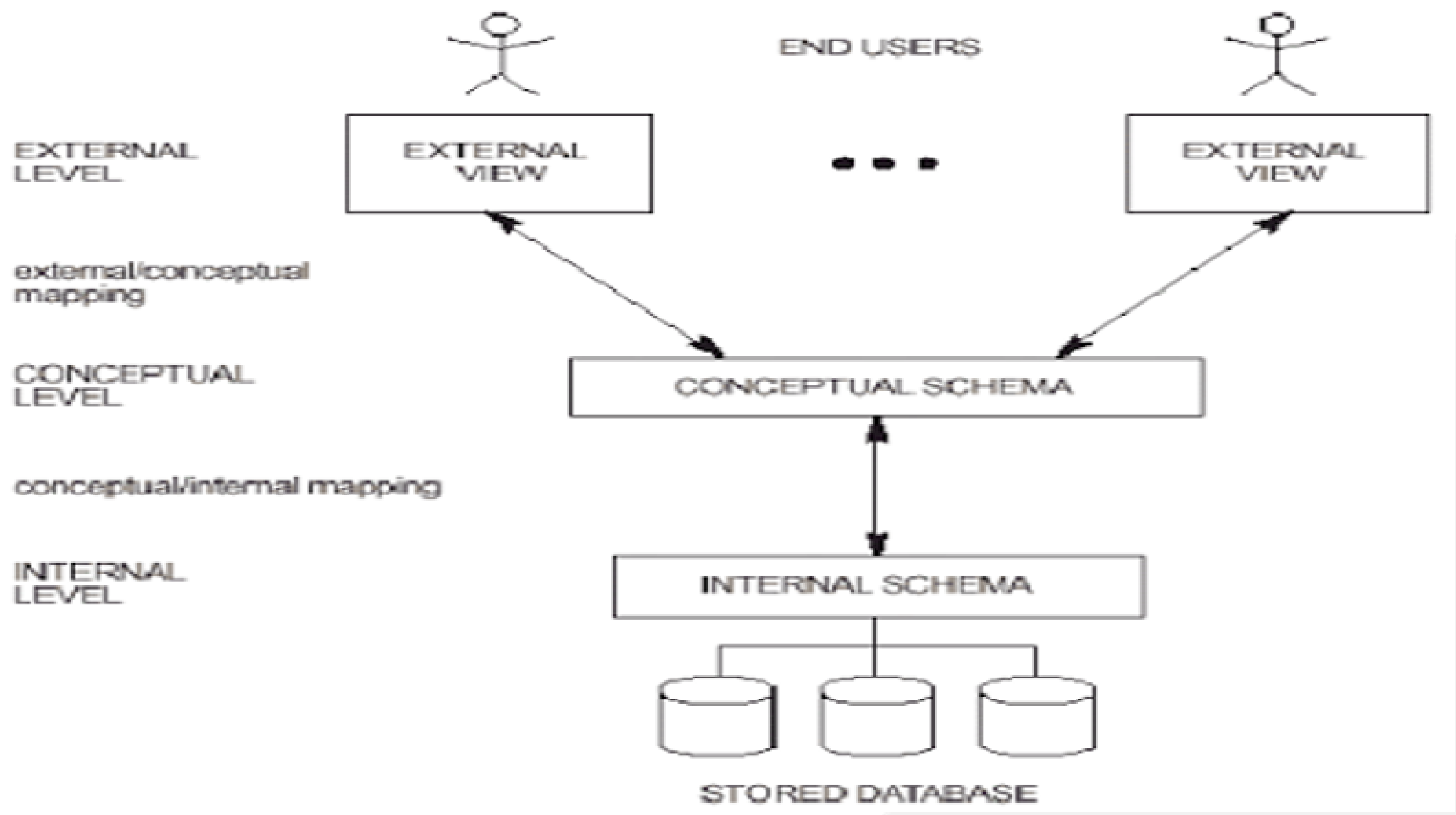


Figure- 1:Three Views of Data (https://in.images.search.yahoo.com/search/images;_ylt=AwrwJTQNqhVd_kAAFe69HAx)

The word **schema** means arrangement – how we want to arrange things that we have to store. The diagram above shows the three different schemas used in DBMS, seen from different levels of abstraction.

Instance – the actual content of the database at a particular point in time

★ Analogous to the value of a variable

Continued

Three General levels :-

Internal View (Physical View) :-

The way the data is stored in the storage media. (Specified by the DBA)

Conceptual View (Logic View):-

Describes the structure and constraints for the whole database. (Specified and used by the programmers).

External View (Sub-Scheme):-

The view of the database as seen by the end user.

Internal or Physical Schema

- The lowest level, called the Internal or Physical schema, deals with the description of how raw data items (like 1, ABC, KOL, H2 etc.) are stored in the physical storage (Hard Disc, CD, Tape Drive etc.).
- It also describes the data type of these data items, the size of the items in the storage media, the location (physical address) of the items in the storage device and so on.
- This schema is useful for database application developers and database administrator.

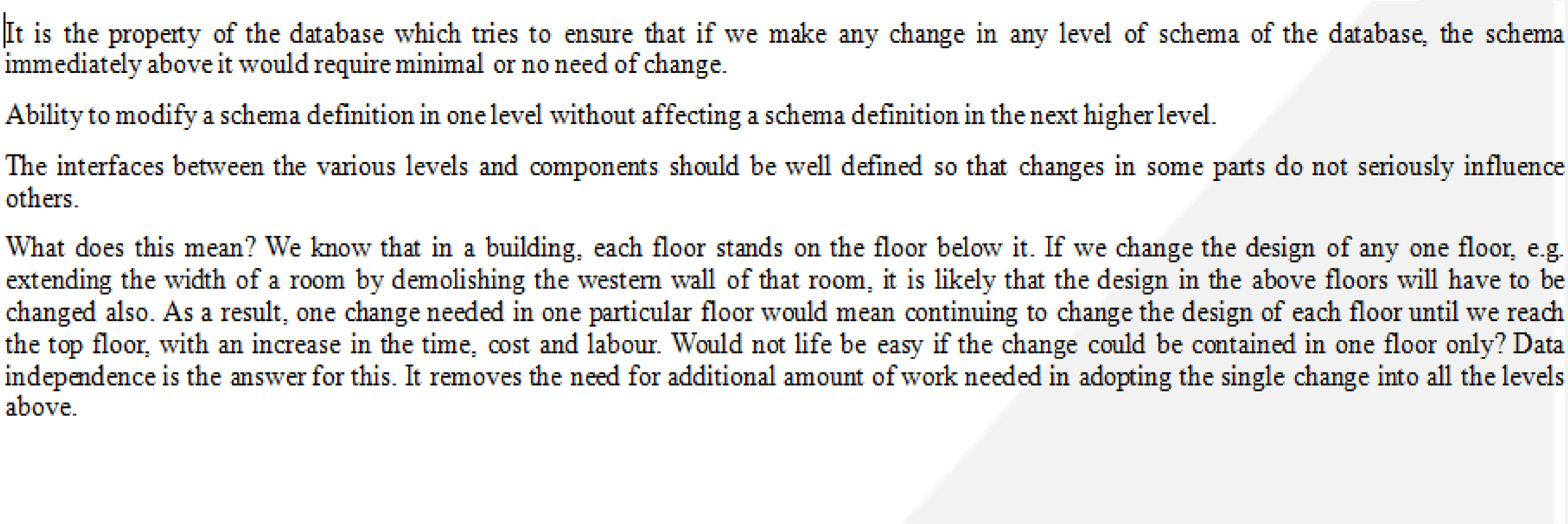
Conceptual or Logical Schema

- The structure of the entire database. Please note that at this level we are not interested with the raw data items anymore, we are interested with the structure of the database.
- This means we want to know the information about the attributes of each table, the common attributes in different tables that help them to be combined, what kind of data can be input into these attributes, and so on.
- Conceptual or Logical schema is very useful for database administrators whose responsibility is to maintain the entire database. The middle level is known as the Conceptual or Logical Schema, and deals with

External or View Schema

- The highest level of abstraction is the External or View Schema. This is targeted for the end users.
- Now, an end user does not need to know everything about the structure of the entire database, rather than the amount of details he/she needs to work with.

Data Independence



It is the property of the database which tries to ensure that if we make any change in any level of schema of the database, the schema immediately above it would require minimal or no need of change.

Ability to modify a schema definition in one level without affecting a schema definition in the next higher level.

The interfaces between the various levels and components should be well defined so that changes in some parts do not seriously influence others.

What does this mean? We know that in a building, each floor stands on the floor below it. If we change the design of any one floor, e.g. extending the width of a room by demolishing the western wall of that room, it is likely that the design in the above floors will have to be changed also. As a result, one change needed in one particular floor would mean continuing to change the design of each floor until we reach the top floor, with an increase in the time, cost and labour. Would not life be easy if the change could be contained in one floor only? Data independence is the answer for this. It removes the need for additional amount of work needed in adopting the single change into all the levels above.

Types of Data Independence

Data independence can be classified into the following two types:

1. Physical Data Independence
2. Logical Data Independence

Physical Data Independence: This means that for any change made in thephysical schema, the need to change the logical schem is minimal. This is practically easier to achieve.

Logical Data Independence: This means that for any change made in the logical schema, the need to change the external schema is minimal. As we shall see, this is a little difficult to achieve.

Database Users

- **Data Base Administrator (DBA):-**
  - Authorizing access to the database
  - Coordinating and monitoring its use
  - Acquiring software and hardware resources
- **Database designers** are responsible for:
  - Identifying the data to be stored
  - Choosing appropriate structures to represent and store this data
- **System analysts**
  - Determine requirements of end users
- **Application programmers**
  - Implement these specifications as programs

Continued

- **End users** : People whose jobs require access to the database
  - Types
    - **Casual end users**: access database occasionally by sophisticated query language when needed.

      (Manager)
    - **Naive or parametric end users**: they make up a large section of the end-user population. Learn only a few facilities that they may use repeatedly

      (bank clerk)
    - **Sophisticated end users**: These include business analysts, scientists, engineers, others thoroughly familiar with the system capabilities.
    - **Standalone users**: Normal users

Database Administrator

- Coordinates all the activities of the database system
  - has a good understanding of the enterprise’s information resources and needs.
- Database administrator's duties include:
  - Schema definition
  - Storage structure and access method definition
  - Schema and physical organization modification
  - Granting users authority to access the database
  - Backing up data
  - Monitoring performance and responding to changes
  - Database tuning

Key Constraint

- Values in a column (or columns) of a relation are unique: at most one row in a relation instance can contain a particular value(s)

- **Key** - set of attributes satisfying key constraint
  - e.g., Id in Student,
  - e.g., (StudId, CrsCode, Semester) in Transcript
- **Super Key**

  Super Key is defined as a set of attributes within a table that uniquely identifies each record within a table. Super Key is a superset of Candidate key.
- **Candidate Key**

  Candidate keys are defined as the set of fields from which primary key can be selected. It is an attribute or set of attribute that can act as a primary key for a table to uniquely identify each record in that table.

Continued

- **Primary Key**

  Primary key is a candidate key that is most appropriate to become main key of the table. It is a key that uniquely identify each record in a table. PK must be **unique** and **Not Null**.

- **Foreign Key**

  Foreign Key is a field or set of fields that are identical to a primary key in another table.

Continued

- **Composite Key**

  Key that consist of two or more attributes that uniquely identify an entity occurance is called **Composite key**. But any attribute that makes up the **Composite key** is not a simple key in its own.

- **Secondary or Alternative key**

  The candidate key which are not selected for primary key are known as secondary keys or alternative keys

- **Non-key Attribute**

  Non-key attributes are attributes other than **candidate key** attributes in a table.

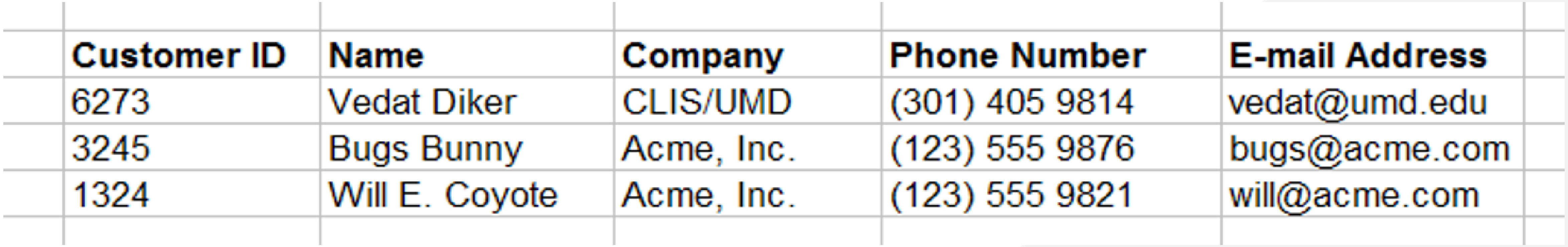
- **Non-prime Attribute**

  Non-prime Attributes are attributes other than **Primary attribute**.

Examples

- Primary Key : CustomerID

Table Name: Customers

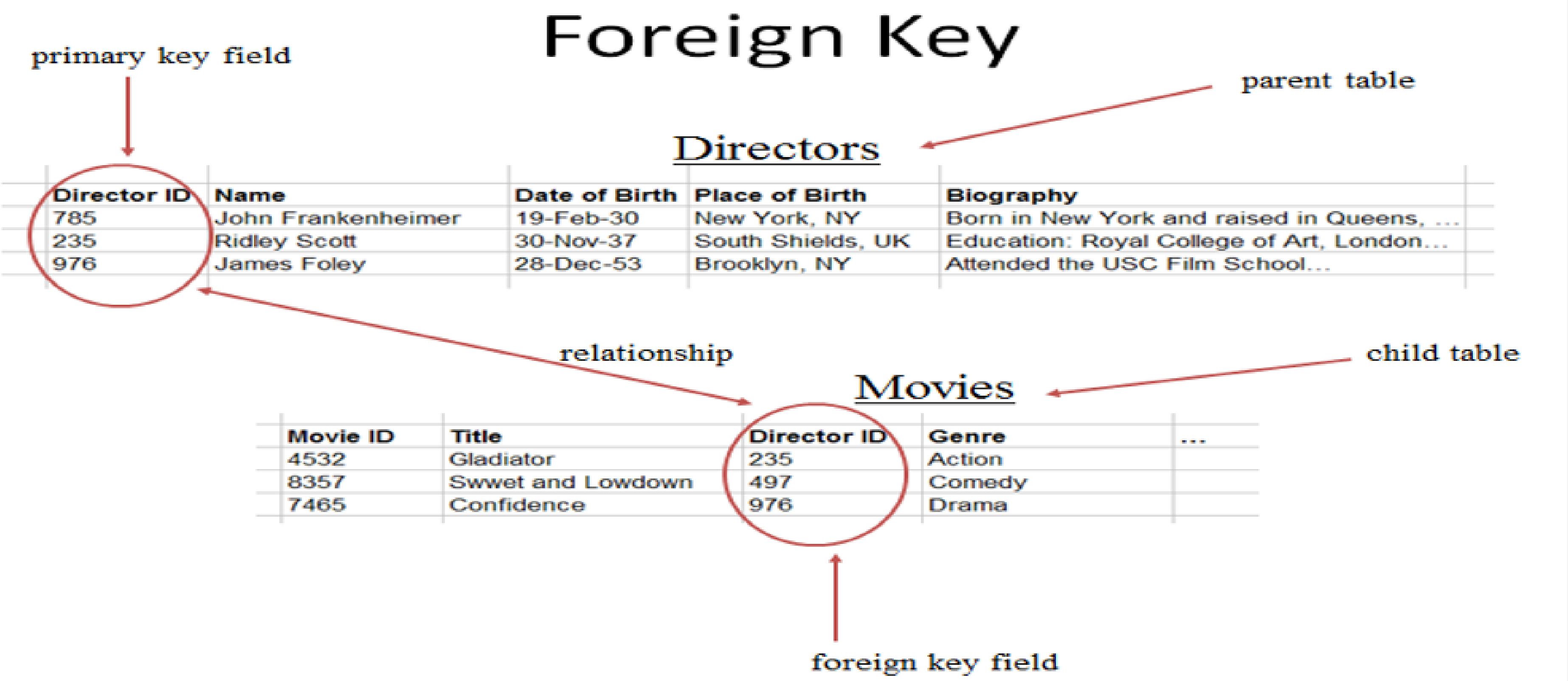


Customer ID	Name	Company	Phone Number	E-mail Address
6273	Vedat Diker	CLIS/UMD	(301) 405 9814	vedat@umd.edu
3245	Bugs Bunny	Acme, Inc.	(123) 555 9876	bugs@acme.com
1324	Will E. Coyote	Acme, Inc.	(123) 555 9821	will@acme.com

Figure : 2:Example of primary Key

(https://in.images.search.yahoo.com/search/images;_ylt=AwrPiFWwqxVdmQMAGgG7HAx)

Primary and Foreign key



Foreign Key

Directors

Director ID	Name	Date of Birth	Place of Birth	Biography
785	John Frankenheimer	19-Feb-30	New York, NY	Born in New York and raised in Queens, ...
235	Ridley Scott	30-Nov-37	South Shields, UK	Education: Royal College of Art, London...
976	James Foley	28-Dec-53	Brooklyn, NY	Attended the USC Film School...

Movies

Movie ID	Title	Director ID	Genre	...
4532	Gladiator	235	Action	
8357	Swwet and Lowdown	497	Comedy	
7465	Confidence	976	Drama	

Figure -3: Examples of Primary and Foreign Key

(https://in.images.search.yahoo.com/search/images;_ylt=AwrPiFWwqxVdmQMAGgG7HAx)

RDBMS

- **Relational Database management System**(RDBMS) is a database management system based on relational model introduced by E.F Codd. In relational model, data is represented in terms of tuples(rows).

- **RDBMS** is used to manage Relational database. **Relational database** is a collection of organized set of tables from which data can be accessed easily. Relational Database is most commonly used database. It consists of number of tables and each table has its own primary key.

Attribute Types

- The set of allowed values for each attribute is called the **domain** of the attribute
- Attribute values are (normally) required to be **atomic**; that is, indivisible
- The special value ***null*** is a member of every domain. Indicated that the value is "unknown"
- The null value causes complications in the definition of many operations

Integrity Constraints

- Domain Constraints
  - Allowable values for an attribute.
  - Domain integrity means the definition of a valid set of values for an attribute. You define
    - data type,
    - length or size
    - is null value allowed
    - is the value unique or not for an attribute.
- Entity Integrity
  - No primary key attribute may be null. All primary key fields **MUST** have data

Entity:- Any thing which has some attributes is called an entity. Like hospital, doctor, car etc

Kinds of Integrity Constraints

- Static - limitation on state of database
    - Syntactic (structural)
        - e.g., all values in a column must be unique
    - Semantic (involve meaning of attributes)
        - e.g., cannot register for more than 18 credits
- Dynamic - limitation on sequence of database states (supported by some DBMSs, but not in current SQL standard)
        - e.g., cannot raise salary by more than 5%
- Tables Operations:- ADD, Delete, Append and Update.
- Integrity Rules :-

✓ PK must be unique.

✓ Related fields should have the same field type.

✓ Related tables should belong to the same DB.

Continued

- **Integrity Conditions**:-

✓ Foreign Key values must be identical to PK values.

✓ Records of primary tables shouldn't be deleted if it is related to another table.

✓ Primary key shouldn't be changed if this record is related to another table.

- **Cascade update related fields :-**

During updating the PK in the primary table the value of the FK should be updated automatically.

- **Cascade deletes related records :-**

During delete a record from the primary table all related records in related tables should be deleted also.

Referential Integrity

The rule that states that any foreign key value (on the relation of the many side) MUST match a primary key value in the relation of

the one side. (Or the foreign key can be null)

- The rules are:

1. You can't delete a record from a primary table if matching records exist in a related table.

2. You can't change a primary key value in the primary table if that record has related records.

3. You can't enter a value in the foreign key field of the related table that doesn't exist in the primary key of the primary table.

4. However, you can enter a Null value in the foreign key, specifying that the records are unrelated.

Referential Integrity

The rule that states that any foreign key value (on the relation of the many side) MUST match a primary key value in the relation of the one side. (Or the foreign key can be null)

- The rules are:

1. You can't delete a record from a primary table if matching records exist in a related table.

2. You can't change a primary key value in the primary table if that record has related records.

3. You can't enter a value in the foreign key field of the related table that doesn't exist in the primary key of the primary table.

4. However, you can enter a Null value in the foreign key, specifying that the records are unrelated.

Entity Integrity Rule

- The entity integrity constraint states that primary keys can't be null.
- There must be a proper value in the primary key field: This is because the primary key value is used to identify individual rows in a table.
- If there were null values for primary keys, it would mean that we could not indentify those rows.
- On the other hand, there can be null values other than primary key fields.
- Null value means that one doesn't know the value for that field. Null value is different from zero value or space.

Assessment Pattern

S.No.	Item	Number/semester	Marks	System
1	MSTs	2	24 (12 each)	Combined tests
2	Quiz	1	4	Once online
3	Surprise test	1	3	Teacher decides
4	Assignments	3 (one per unit)	4	By teacher as per the dates specified
5	Tutorials	Depending on classes	3	In tutorial classes
6	Attendance	Above 90%	2	
Internal (division as mentioned above points 1-6)			**40**	
External			**60**	
Total			**100**	

REFERENCES

- “Data Structures and Algorithms Made Easy: Data Structures and Algorithmic Puzzles” by Narasimha Karumanchi
- “Data Structures Using C” by Aaron M. Tenenbaum
- “Data Structures and Algorithms” by Alfred V. Aho
- “Fundamentals of Data Structures in C by Ellis Horowitz, Sartaj Sahni, and Susan Anderson-Freed

THANK YOU

For queries

Email: Heenae7725@cumail.in