Database
Management System

For

Computer Science
&
Information Technology

By

www.thegateacademy.com
Syllabus for
Data Base Management System

ER-Model, Relational Model, Relational Algebra, Tuple Calculus, SQL, Integrity Constraints, Normal Forms, File Organization, Indexing (E.g., B and B+ Trees), Transactions and Concurrency Control

Analysis of GATE Papers

<table>
<thead>
<tr>
<th>Year</th>
<th>Percentage of Marks</th>
<th>Overall Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>2015</td>
<td>5.00</td>
<td></td>
</tr>
<tr>
<td>2014</td>
<td>8.00</td>
<td></td>
</tr>
<tr>
<td>2013</td>
<td>7.00</td>
<td></td>
</tr>
<tr>
<td>2012</td>
<td>11.00</td>
<td></td>
</tr>
<tr>
<td>2011</td>
<td>7.00</td>
<td></td>
</tr>
<tr>
<td>2010</td>
<td>6.00</td>
<td></td>
</tr>
<tr>
<td>2009</td>
<td>3.33</td>
<td></td>
</tr>
<tr>
<td>2008</td>
<td>7.33</td>
<td></td>
</tr>
<tr>
<td>2007</td>
<td>8.00</td>
<td></td>
</tr>
<tr>
<td>2006</td>
<td>6.00</td>
<td></td>
</tr>
</tbody>
</table>
# Contents

<table>
<thead>
<tr>
<th>Chapters</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>#1. ER Diagrams</strong></td>
<td>1 – 15</td>
</tr>
<tr>
<td>• Levels of Abstraction</td>
<td>1 – 3</td>
</tr>
<tr>
<td>• Entities</td>
<td>3</td>
</tr>
<tr>
<td>• Attribute</td>
<td>4 – 9</td>
</tr>
<tr>
<td>• Specialization</td>
<td>9</td>
</tr>
<tr>
<td>• Utility of E-R Model</td>
<td>10</td>
</tr>
<tr>
<td>• Assignment 1</td>
<td>11 – 12</td>
</tr>
<tr>
<td>• Assignment 2</td>
<td>12 – 13</td>
</tr>
<tr>
<td>• Answer Keys &amp; Explanations</td>
<td>14 – 15</td>
</tr>
<tr>
<td><strong>#2. Functional Dependencies &amp; Normalization</strong></td>
<td>16 – 38</td>
</tr>
<tr>
<td>• Database Application Life Cycle</td>
<td>16 – 18</td>
</tr>
<tr>
<td>• Types of Anomalies</td>
<td>18 – 19</td>
</tr>
<tr>
<td>• The Concept of Functional Dependency</td>
<td>19 – 21</td>
</tr>
<tr>
<td>• Minimal Covers</td>
<td>21</td>
</tr>
<tr>
<td>• Testing Lossless Joins</td>
<td>22 – 24</td>
</tr>
<tr>
<td>• Functional Dependency and the Process of Normalization</td>
<td>25 – 32</td>
</tr>
<tr>
<td>• Assignment 1</td>
<td>33 – 35</td>
</tr>
<tr>
<td>• Assignment 2</td>
<td>35 – 36</td>
</tr>
<tr>
<td>• Answer Keys &amp; Explanations</td>
<td>36 – 38</td>
</tr>
<tr>
<td><strong>#3. Relational Algebra &amp; Relational Calculus</strong></td>
<td>39 – 52</td>
</tr>
<tr>
<td>• Relational Algebra</td>
<td>39 – 40</td>
</tr>
<tr>
<td>• The Projection Operator</td>
<td>40 – 41</td>
</tr>
<tr>
<td>• Relational Algebra Operation from Set Theory</td>
<td>41 – 42</td>
</tr>
<tr>
<td>• Natural Joins</td>
<td>42</td>
</tr>
<tr>
<td>• The Equijoin Operator</td>
<td>42 – 43</td>
</tr>
<tr>
<td>• Outer Joins</td>
<td>44</td>
</tr>
<tr>
<td>• Tuple Relational Calculus</td>
<td>45 – 46</td>
</tr>
<tr>
<td>• The Domain Relation Calculus</td>
<td>46 – 47</td>
</tr>
<tr>
<td>• Assignment 1</td>
<td>48 – 50</td>
</tr>
<tr>
<td>• Assignment 2</td>
<td>50 – 51</td>
</tr>
<tr>
<td>• Answer Keys &amp; Explanations</td>
<td>51 – 52</td>
</tr>
</tbody>
</table>
#4. SQL 53 – 73
- What is SQL 53 – 58
- COLUMN Alias 59 – 62
- Solved Examples 63 – 66
- Assignment 1 67 – 71
- Assignment 2 71 – 72
- Answer Keys & Explanations 72 – 73

#5. Transactions and Concurrency Control 74 – 92
- Transactions 74 – 75
- Serializability 76 – 77
- Concurrency Control Protocols 78 – 80
- Thomas’s Write Rule 81
- Problems of Dirty Data 81
- Multiple Granularity 81 – 83
- Solved Examples 84
- Assignment 85 – 90
- Answer Keys & Explanations 91 – 92

#6. File Structures (Sequential Files, Indexing, B and B⁺ Trees) 93 – 112
- Single Level Ordered Index 93 – 94
- Multilevel Indexes 95 – 96
- Dynamic Multilevel Indexes Using B–Trees and B⁺Trees 97
- Indexes on Multiple Keys 97 – 98
- Search Trees and B-Trees 98 – 109
- Assignment 110 – 111
- Answer Keys & Explanations 112

Module Test 113 – 122
- Test Questions 113 – 119
- Answer Keys & Explanations 120 – 122

Reference Books 123
ER Diagrams

Learning Objectives
After reading this chapter, you will know:
1. Levels of Abstraction
2. Entities
3. Attribute
4. Specialization
5. Utility of E-R Model

DBMS Contains Information about a Particular Enterprise
- Collection of interrelated data
- Set of programs to access the data
- An environment that is both convenient and efficient to use

Levels of Abstraction

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.
### Points to Emphasize
- How to use Entity-Relationship (ER) modeling in database design.
- The basic concepts associated with the Entity-Relationship (ER) model.
- A diagrammatic technique for displaying an ER model using the Unified Modeling Language (UML).
- How to identify and resolve problems with ER models called connection traps.
- How to build an ER model from a requirements specification.

### What to Model?
#### Static Information
- Data -- Entities
- Associations -- Relationships among entities

#### Dynamic Information
- Processes -- Operations/Transactions
- Integrity constraints -- Business Rules/Regulations and Data meanings

### What is data Model?
A collection of tools for describing:-
- Data
- Data relationships
- Data semantics
- Data constraints

**Data Model**: A data model is a collection of concept that can be used to describe the structure of database.

**Schema**: The description of a database is called the database schema.

### System Model Tools
- Data flow diagram (DFD)
- Hierarchical input process and output (HIPO)
- State transition diagrams (STD)
- Entity Relationship (ER) diagrams

### Entity-Relationship Model (ER Model): A data model in which information stored in the database is viewed as sets of entities and sets of relationships among entities and it is diagram-based representation of domain knowledge, data properties etc..., but it is more intuitive and less mechanical. Entity – Relationship is a popular high-level conceptual data model.
**ER Diagram Symbols Figure**

- **E**: Entity Set
- **R**: Relationship Set
- **A**: Attribute
  - **Multi Valued Attribute**
  - **Derived**
  - **Key**
  - **Total participation of Entity Set in Relationship**
  - **Discriminating Attribute of Weak Entity Set**

**Components of E-R Model**

(a) Entity
(b) Relationship
(c) Attributes

**Entities**

The basic object that the ER model represents is an entity, which is a “thing” in the real world with an independent existence and is distinguishable.

**Example:** Student entities with unique roll number
Account entities with unique account number

**Example:** Identify two entities that might be important for a retail business.
**Solution:** Two entities for a business might include employee and customer.

**Entity Set**
A set of entities of the same type.

**Example:** All the student entities in a college
Entity sets need not be disjoint.

**Example:** A person entity could be in both the customer and employee sets

**Types of Entities**

- Entities with physical existence
  - Example: Student, customer, book etc
- Entities with Conceptual existence
  - Example: Sale, University course etc

**Relationship**
An association among two or more entities.

**Example:** The relationship between a Faculty and Student i.e. faculty take course for student

**Relationship Set**
A set of relationships of the same type
Attribute

The particular properties of entity that describe it.

Example: A student entity might have attributes such as: Roll number, Name, Age, Address etc.

As all entities in an entity set have the same attributes, entity sets also have attributes - the attributes of the contained entities. The value of the attribute can be different for each entity in the set.

In ER Model Attributes can be Classified into the Following Types

- Simple and Composite Attribute
- Single Valued and Multi Valued attribute
- Stored and Derived Attributes

Simple and Composite Attribute

Simple attribute that consist of a single atomic value. A composite attribute is an attribute that can be further subdivided. For example the attribute ADDRESS can be subdivided into street, city, state, and zip code. A simple attribute cannot be subdivided. For example the attributes age, sex etc are simple attributes.

Simple Attribute: Attribute that consist of a single atomic value.

Example: Salary, Age etc

Composite Attribute: Attribute value not atomic.

Example: Address: ‘House no: City: State’
Name: ‘First Name: Middle Name: Last Name’

Single Valued and Multi Valued Attribute

A single valued attribute can have only a single value. For example a person can have only one 'date of birth', 'age' etc. That is a single valued attributes can have only single value. But it can be simple or composite attribute. That is 'date of birth' is a composite attribute, 'age' is a simple attribute. But both are single valued attributes. Multivalued attributes can have multiple values. For instance a person may have multiple phone numbers, multiple degrees etc. Multivalued attributes are shown by a double line connecting to the entity in the ER diagram.

Single Valued Attribute: Attribute that hold a single value
Example1: Age
Example2: City
Example3: Customer id

Multi Valued Attribute: Attribute that hold multiple values.
Example1: A customer can have multiple phone numbers, email id's etc
Example2: A person may have several college degrees

Stored and Derived Attributes

The value for the derived attribute is derived from the stored attribute.

For example 'Date of birth' of a person is a stored attribute. The value for the attribute 'AGE' can be derived by subtracting the 'Date of Birth'(DOB) from the current date. Stored attribute supplies a value to the related attribute.

Stored Attribute: An attribute that supplies a value to the related attribute.
Example: Date of Birth
Derived Attribute: An attribute that's value is derived from a stored attribute.
Example: Age and its value is derived from the stored attribute Date of Birth.

**Domain of Attribute:** A set of possible values for an attribute (the type of the attribute).
Example: The domain of student name might be strings of some fixed length.
The domain of roll number might be 10 digit positive integers or alphanumeric.

**Relationship Degrees**

<table>
<thead>
<tr>
<th>Connectivity</th>
<th>Diagram</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>One – to – One</td>
<td><img src="image" alt="One-to-One Relationship" /></td>
<td>The one-to-one relationship has the cardinality or degree of one and only one in both directions.</td>
</tr>
<tr>
<td>One – to - Many</td>
<td><img src="image" alt="One-to-Many Relationship" /></td>
<td>The many-to-one relationship has cardinality in one direction of one or more and in the other direction of one and only one.</td>
</tr>
<tr>
<td>Many – to - Many</td>
<td><img src="image" alt="Many-to-Many Relationship" /></td>
<td></td>
</tr>
</tbody>
</table>

Supervised by
![Supervised by](image)

Managed by
![Managed by](image)

Has
![Has](image)

Works-on
![Works-on](image)

Start-date
![Start-date](image)

Task-Assignment
![Task-Assignment](image)
Types of Relationships

Example: Indicate how you would read the following relationship. Identify the type of relationship.

Solution:

L to R: Each Car responsibility must be given to one and only one driver.
R to L: Each DRIVER must be responsible for one or more CARS.
This is a M: One relationship, mandatory in both directions.

Multiplicity: Multiplicity constraints the way that entities are related - it is a representation of the policies (or business rules) established by the user or enterprise. Multiplicity actually consists of two separate constraints.

Cardinality: Cardinality describes the maximum number of possible relationship occurrences for an entity participating in a given relationship type i.e. how many relationship instances is an entity permitted to be linked to.

Participation: Participation determines whether all or only some entity occurrences participate in a relationship i.e., how is an entity linked to the relationship.

Total Participation (indicated by double line): Every entity in the entity set participates in at least one relationship in the relationship set.

Partial Participation: Some entities may not participate in any relationship in the relationship set.

Note: Cardinality limits can also express participation constraints.