Database Management System – MC212 – Lab Manual

Objective: For a case study, design and implement database using relational database PostgreSQL with efficient information storage and retrieval. Perform database operations using SQL client pgAdmin and through Java client utility to provide answer to complex questions from the data in database.

Database Management System – MC212

Lab 1

Develop Problem Description of case study

- Decide on the case study by learning different domains form resources such as internet e.g. insurance, finance, product management, medical science, project management. This is known as gaining business domain knowledge.
- Create a Document as Lab1_IDNos_ProjectDescription_VersionNo.pdf (e.g. Lab1_StudIDs_ProjectDescription_v1.pdf) with following information.
    - Describe the Problem Statement or Scope of your case study.
    - Articulate information about the database requirements of this case study.
    - Provides expected list of questions (queries) that can be answered from this database once designed.

This is a living document. You can enhance your case study by adding additional requirements and questions or queries during subsequent labs.

Database Management System – MC212

Lab 2

Finalize the Problem Description and Database Requirements of case study

- Finalize the Problem Description
- Finalize the Database Requirements as below
    - You should have at least 10 Entities Identified which will in turn be represented as 10 requirements. 10 entities can include any weak entity that you may have.
    - Determine the relationships between each of the entities. Create requirements representing each relationship between entities. Ensure that each requirement is detailed enough to determine the cardinality and participation constraints for Entity Relationship Model.
    - Relationship requirements should be engaging maximum 2 or 3 entities only for binary or ternary relationships respectively. You should also include requirements for any recursive relations if you have in your project.
    - You should have at least 20-30 detailed database requirements once you document all the entities and relationships.
- Finalize the Questions/Queries
    - From the proper documented database requirements, you should be able to determine what kind questions can be answered from the project database.
    - Questions should be in terms of providing answer to statistical information from the database.
    - Adding or deleting data to/from the database cannot be counted in these queries.
- Create a new version of document as Lab2_IDNos_ProjectDescription_VersionNo.pdf (e.g. Lab2_StudIDs_ProjectDescription_v1.pdf) with following information.

Database Management System – MC212

Lab 3

E-R Model

You can use open source Dia tool or any other flowcharting tool that you may be familiar with to draw the ER Model.

PART A: Produce ERD version 0 with following details

- Identify Entity Types, Relationship Types, Entity Sets, Relationship Sets, and Attributes for each of them by Noun Analysis and Verb Analysis.

Part A need to be submitted in File: Lab3_StudIDs_ERD_v0.pdf

PART B: Produce ERD version 1 with following details

- Identify Weak Entity if any
- Identify Keys for each entity type.
- Identify cardinality constraint (1 to 1, 1 to many, many to 1 and many to many).
- Identify Participation (total or partial) constraints.
- Identify 'IS A' hierarchy if any. Indicate if 'IS A' hierarchy has overlapping or disjoint constraints.
- Identify if you have any Unions between entity types.

Part B need to be submitted in File: Lab3_StudIDs_ERD_v1.pdf

Note: Both PDF files should be submitted in a single zip file Lab3_StudIDs_ERD.zip

Database Management System – MC212

Lab 4

E-R Model to Relational Model

In this Lab, you will develop relational model from ER model with the mapping rules we learned in the class.

- Map all strong entity types to relations; handle Multi-value attributes and composite attributes.
- Map weak entity type to relations considering key attributes of owner entity types as foreign key.
- Map binary or ternary relationships with cardinality and participation constraints.
    - Cardinality constraint of one-to-one relationship is mapped by using primary key of total participation entity into foreign key attribute of partial participation entity
    - Cardinality constraint of one-to-many relationship is mapped by using primary key of entity with a one tuple into foreign key attribute of entity with many tuples
    - Cardinality constraint of many-to-many relationship is mapped by creating a separate relation of the relationship in which the primary key of the new relation will be combination of primary keys of both participating entities.
- Map any IS-A or Union relationship with rules of overlapping or disjoint constraints

Relational Model to be submitted in File: Lab4_StudIDs_RelationalModel_v1.pdf

Please submit the final versions of Requirements Document and ER Model with latest changes in new version number. You should submit one zip file with requirements documents, ER model and Relational Model Lab4_StudIDs_RelationalModel.zip

Database Management System – MC212

Lab 5

Relational Model to SQL DDL

In this Lab, you will develop SQL Data Definition Language (DDL) statements using the mapping rules from Relations Model to DDL.

- Create your own schema(s) as required
- Create tables in schema for each mapped relation in relational model. Pay attention to the following while creating tables
    - Use of domain i.e. data type for each attribute
    - Domain constraints (Note: you may use check constrain for specific values or range)
    - Key constraints (Primary Keys)
    - Referential Integrity Constraints (Reference Schemas and Foreign Keys)
        - DDL should include violation handling for insert, delete or update using different options (SET NULL, CASCADE, RESTRICT, SET DEFAULT etc)
- Identify any other constraints (such as application specific constraint) that cannot be implemented in DDL and has to handled at application level or in SQL code

SQL DDL to be submitted in File: Lab5_StudIDs_DDL_v1.pdf

Database Management System – MC212

Lab 6 and Lab 7

Normalization and Schema Refinement

- Normalization & Schema Refinement
  - Generate functional dependencies for every schema
  - Determine whether the schema is in 1NF, 2NF, 3NF or BCNF form.
  - Normalize it further to 3NF/BCNF from 1NF or 2NF. If relations are NOT 3NF or lower form and cannot be normalized into BCNF, please provide a reason for not being able to normalize further.
  - Create tables using normalized design (removing your original version of relations)
  - Define constraints of all types (domain, PK, FK, Referential) for these tables
- Submissions
  - **Lab6_StudIDs_FD_v1.pdf or Lab7_StudIDs_FD_v1.pdf**:
    - List update, delete, and insert anomalies in original database design. scrutiny/discussion of various types of redundancies must also be included
    - Document the logic of how you arrived at the 3NF/BCNF design step by step
  - **Lab6_StudIDs_DDL_v1.pdf or Lab7_StudIDs_DDL_v1.pdf**:
    - Normalized DDL Statements: Create table statements for all relations which are there in new design, with appropriate constraints defined. If no change is required, state the reason.

Database Management System – MC212

Lab 8, 9 and 10

Relation Algebra and SQL Queries for Functional Requirements

The goal for this assignment to have a working database that satisfies all the functional requirements you must have submitted earlier.

- Finalize the requirements and ensure that your database structure can support all the query requirements. If needed please make appropriate changes. (**Note: You need to write queries to provide answers to each of the requirements**)
- Create Tables and Views (if any) by executing DDLs.
- Populate the database tables using INSERT INTO statement, with **each table having sufficient number of records to be able to get good results from queries**.
- Write queries in **Relational Algebra and SQL** to satisfy each of the requirements for your respective project. **You should have at least 20 queries**. Queries should be able to answer wide range of questions and should consist of following types
    - Basic queries with conditions and joins
    - Queries using aggregation in SELECT and with aggregated conditions in HAVING clause
    - Queries having nested queries
    - Queries having aggregation in nested queries
    - Correlated queries
    - Queries with division operation
- Run the SQL queries onto the database instance and capture the output.

Submission:

1. Updated Requirements Document if changed.

2. Updated DDLs if changed.

3. **Lab7_StudIDs_Queries_v1.pdf: In this document for each of the requirements you need to:**

    1. **Write the requirement with requirement no**
    2. **Write the query in Relational Algebra that provide answer to question in requirement**
    3. **Write equivalent SQL query**
    4. **Capture output from the SQL query**

You should submit one zip file with all the modified documents Lab8_StudIDs.zip or Lab9_StudIDs.zip. Please submit the final versions of all documents if changed with a new version number in Lab 10 as Lab10_StudIDs.zip

Database Management System – MC212

Lab 11

Programming – Stored Procedures (Functions), Triggers, Cursors, Views

- You may have various roles (admin, general user, guest etc) supporting different database content access. Create Views (if any) for each of these roles to access only the required content for the role.
- Implement stored procedures/functions and/or cursors to provide functional requirements requiring complex queries or multi-step queries.
- See if you have any requirement that require you to use triggers and implement them.

Submission:

1. Updated Requirements Document if changed because of additional requirements established using programming constructs.

2. **Lab11_StudIDs_prog_v1.pdf**: DDLs for views, stored procedures/functions and triggers.

3. Captured output from execution of triggers and stored procedures.

You should submit one zip file with all the modified documents Lab11_StudIDs.zip.

Database Management System – MC212

Lab 12

Programming – JDBC API

- Implement Core Java code to use JDBC API for executing the queries of following nature:
    - Data retrieval using SELECT query
    - Data modification using UPDATE query
    - Add new data using INSERT query
    - Remove the rows using DELETE query
    - Calling Stored Procedure
- During the implementation of the above functionalities, you need to show how to use Connection, Statement, PreparedStatement, ResultSet and CallableStatement classes from JDBC API.

Submission:

1. Updated Requirements Document if changed because of additional requirements established using programming constructs.

2. **Lab12_StudIDs_prog_v1.java source file for java code**

3. Captured output from execution of java code.

You should submit one zip file with all the modified documents Lab12_StudIDs.zip.