

# **BA ACADEMY 2025**

## **SQL Capstone Project (A) BA - B1**

### **PROJECTS REPORT**

**GITHUB Link:**

<https://github.com/tarangkashyaptk/BAacademySQLProject-TarangKashyap>

**TARANG KASHYAP**  
**katarang@amazon.com**

# Table of Contents

1. Introduction
2. Objective
3. Learning
4. Technologies/Tools Used
5. Project 1: Academic Management System
6. Project 2: Student Database Management System
7. Project 3: Event Management System
8. Project 4: OLAP Operations

## **Introduction:**

This report presents the combined work for four structured SQL projects covering practical database design, normalization, data manipulation, reporting, and OLAP operations. Each project demonstrates hands-on application of database concepts and SQL programming to solve real-world data management scenarios.

## **Objective:**

The objective is to design, implement, and demonstrate various database management systems and analytical solutions using SQL, applying concepts such as table design, relationships, constraints, data insertion, CRUD operations, and OLAP techniques.

## **Learning:**

- Understanding of relational database design principles
- Use of Primary Key and Foreign Key constraints
- Data insertion, updates, deletions, and integrity constraints
- Use of aggregate functions, grouping, sorting, and joins
- Application of OLAP operations (Drill Down, Rollup, Cube, Slice, Dice)
- Practical SQL coding in MySQL and PostgreSQL

## **Technologies/Tools Used:**

- MySQL
- PostgreSQL
- SQL syntax and query editors (pgAdmin, MySQL Workbench)
- ER Diagrams (Draw.io or Lucidchart)

## ❖ Project 1: Academic Management System

### Project Scope:

Design a system to manage student information and course enrollment with three tables: StudentInfo, CoursesInfo, and EnrollmentInfo.

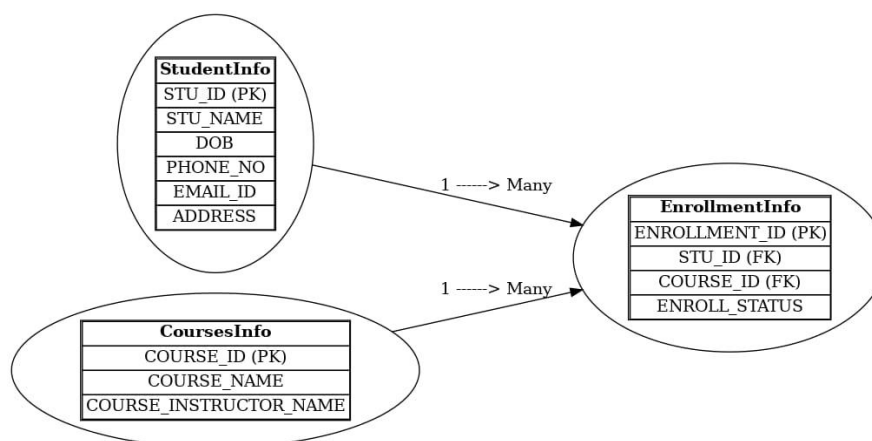
### Database Design:

- StudentInfo: STU\_ID, STU\_NAME, DOB, PHONE\_NO, EMAIL\_ID, ADDRESS
- CoursesInfo: COURSE\_ID, COURSE\_NAME, COURSE\_INSTRUCTOR
- EnrollmentInfo: ENROLLMENT\_ID, STU\_ID (FK), COURSE\_ID (FK), ENROLL\_STATUS

### Table Structure:

Normalized to handle student data, courses, and enrollment status with referential integrity.

### Entity Relationship Diagram:



## **Data Insertion:**

Sample records were inserted in each table to simulate real enrollments.

## **Queries Executed:**

- Retrieve student details and enrollment status
- Retrieve list of courses for a student
- Retrieve course details
- Join queries for reports (number of students per course, per instructor, multiple enrollments)

## **SQL Concepts Demonstrated:**

- CREATE, INSERT, SELECT, JOIN, GROUP BY
- FOREIGN KEY constraints
- Subqueries and conditional statements

## **Expected Output:**

Accurate student-course enrollment reports.

## **How to Execute:**

Run CREATE TABLE scripts, insert data, execute queries.

## **Conclusion:**

The Academic Management System successfully demonstrates relational design and reporting.

## **GITHUB Link:**

<https://github.com/tarangkashyaptk/BAacademySQLProject-TarangKashyap>

## ❖ Project 2: Student Database Management System

### Project Scope:

Create a complete student database for storing detailed student records with validation.

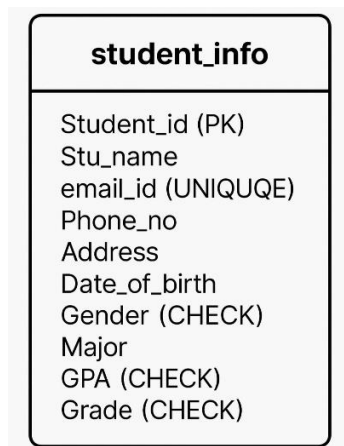
### Database Design:

student\_info table with: Student\_id, Stu\_name, Department, email\_id, Phone\_no, Address, Date\_of\_birth, Gender, Major, GPA, Grade.

### Table Structure:

Single-table design with unique constraints and checks on Gender, GPA, and Grade.

### Entity Relationship Diagram:



### Data Insertion:

10 diverse student records.

## Queries Executed:

- Retrieve all students ordered by grade
- Retrieve male students
- Retrieve students with GPA < 5
- Update student email and grade
- Group by Department & Gender
- Rename table
- Retrieve student with highest GPA

## SQL Concepts Demonstrated:

- Constraints: PRIMARY KEY, UNIQUE, CHECK
- SELECT, UPDATE, GROUP BY, ORDER BY, ALTER TABLE

## Expected Output:

Clean student information with integrity.

## How to Execute:

Create the database, use USE student\_database, run table creation, insert data, and execute queries.

## Conclusion:

This project builds solid practice in table creation and data integrity enforcement.

## GITHUB Link:

<https://github.com/tarangkashyaptk/BAacademySQLProject-TarangKashyap>

## ❖ Project 3: Event Management System

### Project Scope:

Develop a system to manage events, attendees, and registrations.

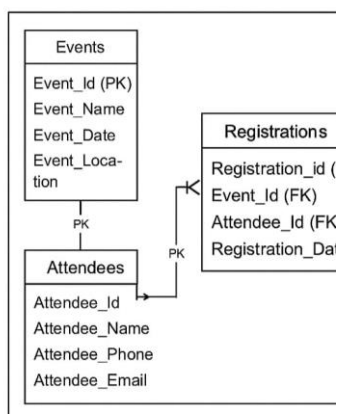
### Database Design:

- Events: Event\_Id, Event\_Name, Event\_Date, Event\_Location, Event\_Description
- Attendees: Attendee\_Id, Attendee\_Name, Attendee\_Phone, Attendee\_Email, Attendee\_City
- Registrations: Registration\_id, Event\_Id (FK), Attendee\_Id (FK), Registration\_Date, Registration\_Amount

### Table Structure:

Proper normalized structure with foreign keys linking attendees to events.

### Entity Relationship Diagram:





## **Data Insertion:**

Sample events, attendees, and registrations.

## **Queries Executed:**

- Insert, update, delete events
- Add new attendees
- Register attendees
- Reporting: attendee lists, total attendees per event, total revenue per event

## **SQL Concepts Demonstrated:**

Joins, GROUP BY, SUM, COUNT, DELETE, UPDATE

## **Expected Output:**

Event reports showing registrations and revenue.

## **How to Execute:**

Create the EventsManagement database, tables, insert data, and run queries.

## **Conclusion:**

The Event Management System illustrates relationship handling and reporting for event organizers.

## **GITHUB Link:**

<https://github.com/tarangkashyaptk/BAacademySQLProject-TarangKashyap>

## ❖ Project 4: OLAP Operations

### Project Scope:

Perform OLAP analysis to understand sales data using Drill Down, Rollup, Cube, Slice, and Dice operations.

### Database Design:

sales\_sample table: Product\_Id, Region, Sales\_Date, Sales\_Amount, Product\_Name.

### Table Structure:

Simple sales dataset with multiple dimensions: Region, Product, Date.

### Entity Relationship Diagram:

<b>sales_sample</b>
Product_Id (PK)
Region
Sales_Date
Sales_Amount
Product_Name

### Data Insertion:

10 sales records with varied regions, products, and dates.

### Queries Executed:

- Drill Down: Region → Product level
- Rollup: Product → Region → Grand Total
- Cube: All possible combinations of Region, Product, Date
- Slice: Filter by Region or Date Range
- Dice: Filter by Region + Product + Date Range

## SQL Concepts Demonstrated:

- GROUP BY with ROLLUP and CUBE
- Aggregate functions (SUM)
- WHERE filtering for Slice and Dice

## Expected Output:

Detailed sales insights across dimensions.

## How to Execute:

Create the table, insert data, run each OLAP query to check summaries and breakdowns.

## Conclusion:

This project demonstrates OLAP capabilities for advanced sales data analysis.

## GITHUB Link:

<https://github.com/tarangkashyaptk/BAacademySQLProject-TarangKashyap>

X-----X