Laboratory Work #3

3.1 Please write SQL queries for following tasks and save as .sql file.

1. Create a database called lab3.
2. Download and run the lab3.sql file using Query Tool (make sure the tables are created correctly).
3. SELECT the last name of all students.
4. SELECT the last name of all students, without duplicates.
5. SELECT all data of students whose last name is "Johnson."
6. SELECT all data of students whose last name is "Johnson" or "Smith."
7. SELECT all data of students who are registered in the "CS101" course.
8. SELECT all data of students who are registered in the "MATH201" or "PHYS301" courses.
9. SELECT the total number of credits for all courses.
10. SELECT the number of students registered for each course. Show the course ID and the number of students. (Use the COUNT(*) operator for counting the number of students.)
11. SELECT the course ID with more than 2 students registered.
12. SELECT the course name with the second-highest number of credits.
13. SELECT the first and last names of students registered in the course with the fewest credits.
14. SELECT the first and last names of all students from Almaty.
15. SELECT all courses with more than 3 credits, sorted by increasing credits and decreasing course ID.
16. Decrease the number of credits for the course with the fewest credits by 1.
17. Reassign all students from the "MATH201" course to the "CS101" course.
18. Delete from the table all students registered for the "CS101" course.
19. Delete all students from the database.

3.2 Database Systems. Relational Model & Keys

Tasks:

Tools Required: ERDPlus, Draw.io (or similar ER tool), DB Fiddle, pen and paper

This assignment provides hands-on experience with relational database fundamentals including key identification, ER modeling, schema mapping, and normalization techniques. You will work through progressively complex scenarios that mirror real-world database design challenges.

Part 1: Key Identification Exercises

Task 1.1: Superkey and Candidate Key Analysis

Relation A: Staff

Staff(StaffID, NationalID, WorkEmail, Mobile, FullName, Unit, AnnualSalary)

Sample Data:

StaffID	NationalID	WorkEmail	Mobile	FullName	Unit	AnnualSalary
201	555-66-7777	alice@uni.edu	770-2001	Alice Turner	Research	82000
202	111-22-3333	robert@uni.edu	770-2002	Robert Kim	Finance	76000
203	888-99-0000	emma@uni.edu	770-2003	Emma Novak	Research	80000

Your Tasks:

1. List at least **4 different superkeys** for the Staff relation (show the attribute sets).
2. Identify **all candidate keys**.
3. Which candidate key would you choose as the **primary key** and why? Provide justification (uniqueness, stability, privacy, etc.).
4. Could two staff members share the same mobile number? Explain whether the given data supports that and what business rule you would choose.

Relation B: CourseEnrollment

CourseEnrollment(StudentNo, SubjectCode, GroupNo, Term, Year, Grade, CreditHours)

Business Rules:

- A student may take the same subject in different terms/years.
- A student cannot enroll in the same SubjectCode & GroupNo combination in the same Term & Year twice.
- Each Group in a term has a fixed CreditHours value.

Your Tasks:

1. Determine the **minimum set of attributes** required for the primary key. Explicitly list them.
2. Explain why each attribute in that key is necessary (tie to the business rules).
3. Identify any other **candidate keys** (if any) and explain why they are or are not valid.

Task 1.2: Foreign Key Design

Given these tables for the university:

- Student(StudentNo, FullName, Email, MajorCode, AcademicAdvisorID)
- Lecturer(LecturerID, FullName, DeptCode, Title)
- Module(ModuleID, Title, Credits, DeptCode)
- Faculty(DeptCode, DeptName, DeanID)

- Registration(StudentNo, ModuleID, Term, Year, Status)

Your Tasks:

1. Identify all **foreign key relationships** between these tables. For each FK state: referencing table, referencing attribute(s), referenced table, referenced attribute(s), and whether the FK should be **ON DELETE CASCADE**, **SET NULL**, or **RESTRICT**, with justification.

Part 2: ER Diagram Construction
Task 2.1: Clinic Information System
Scenario Requirements:

- Patients: patient numbers (unique), name, date of birth, contact addresses (may have multiple), multiple phone numbers, emergency contact, insurance provider and policy number.
- Physicians: physician ID (unique), name, specialties (physician may have multiple specialties), contact numbers, office room(s), and employment start date.
- Clinics/Departments: code, name, floor location.
- Appointments: record patient — physician meetings with appointment datetime, reason, duration (in minutes), and visit notes.
- Treatments: each treatment entry links to an appointment and may include procedure codes, costs, and follow-up instructions.
- Rooms: rooms are identified by (ClinicCode, RoomNumber) — room numbers are reused across clinics.

Your Tasks:

1. List all entities and indicate which are **strong** and which (if any) are **weak**.
2. For each entity list attributes and classify them as **simple, composite, multi-valued, or derived**.
3. Identify relationships and state cardinalities (1:1, 1:N, M:N); note any associative entities required.
4. Draw a complete **ER diagram** using Chen or Crow's Foot notation; mark primary keys clearly.
5. Suggest how to handle physicians with multiple specialties (one-to-many vs. M:N with a junction table) and justify your choice.

Task 2.2: Online Marketplace
Scenario Requirements:

- Customers place Orders. Orders contain OrderItems (each item records quantity and price at purchase time).
- Products belong to Categories (a product can belong to multiple categories) and are provided by Suppliers.
- Customers can have multiple shipping addresses distinct from billing address.
- Products have CustomerReviews (rating, review text, review date) — a customer may review a product only once per order of that product.
- Inventory tracks stock by Warehouse and SKU; warehouses are physical locations with address info.

Your Tasks:

1. Provide a full **ER diagram** capturing Customers, Orders, OrderItems, Products, Categories, Suppliers, Warehouses, Inventory, and Reviews.
2. Identify a **weak entity** in this design (if any) and explain why it is weak.

3. Identify at least one **many-to-many relationship that requires attributes** (e.g., Product–Category or Product–Warehouse inventory snapshot), and show how you model it (associative entity with attributes).

Part 4: Normalization Workshop
Task 4.1: Denormalized Table Analysis
Given Table:
ResearchParticipation(StudentNo, StudentName, Major, ProjectCode, ProjectTitle, ProjectDomain, SupervisorID, SupervisorName, SupervisorDept, RoleInProject, HoursContributed, StartDate, EndDate)
Your Tasks:

1. **List all functional dependencies (FDs)** you can infer from the attributes (use the format A → B).
2. **Identify redundancy:** describe what information repeats and provide concrete examples.
3. **Anomalies:** give examples of update, insert, and delete anomalies that would arise.
4. **1NF:** Are there 1NF violations? If yes, fix them and show the corrected schema.
5. **2NF:** Propose the primary key for the denormalized table; detect any partial dependencies; provide a 2NF decomposition.
6. **3NF:** Identify transitive dependencies and provide a final 3NF decomposition with table schemas and keys.

Task 4.2: Advanced Normalization
Given Table:
ClassTimetable(StudentNo, StudentMajor, CourseCode, CourseTitle, InstructorID, InstructorName, Slot, RoomNumber, Campus)
Business Rules:

- Each student has one declared major.
- Each course code maps to exactly one course title.
- Instructors are identified by InstructorID and have a fixed name.
- Slot + RoomNumber together imply Campus (rooms are unique per campus but may repeat numbers across campuses).
- Each course section (CourseCode taught at a Slot in a Room) has exactly one instructor.
- Students can enroll in many course sections.

Your Tasks:

1. Determine the correct **primary key** for the table (explain your reasoning).
2. List all **functional dependencies**.
3. Is the table in **BCNF**? If not, decompose it to BCNF, showing every step and the schemas that result.
4. Explain any potential **loss of information** or need for joins after decomposition.

Part 5: Design Challenge — Campus Organizations
Scenario Requirements:

- Track student clubs and societies, memberships, event scheduling, officer positions, faculty advisors, room bookings, and budgets.
- Students may belong to multiple clubs; clubs have many members.

- Clubs run events; events have attendees and may reserve rooms (room reservations include start/end times and expected attendance).
- Clubs have officers (a student holding a role for a given term), and each club has exactly one faculty advisor (faculty may advise multiple clubs).
- Budget tracking: clubs have budgets and record expenses (expense date, amount, purpose, approved-by).

Your Tasks:

1. Create a full **ER diagram** for the system (entities, relationships, cardinalities, and keys).
2. Convert the ER model to a **normalized relational schema** (at least to 3NF). Include primary keys, foreign keys, and suggested data types.
3. Identify one design decision with multiple valid alternatives (e.g., modeling officers as attributes vs. as a relationship table) and **justify** your chosen approach.
4. Write **three example queries** the system must support (in plain English), for example: – "Find all students who are officers in the Engineering Society" – "List all upcoming club events with room reservations and expected attendance"

Lab Deliverables

What to Submit:

1. **Complete Solutions:** Detailed answers to all tasks with clear explanations
2. **ER Diagrams:** Hand-drawn or digital diagrams for Tasks 2.1, 2.2, and 5.1
3. **Normalization Work:** Step-by-step decomposition showing all intermediate steps
4. **Relational Schemas:** Complete table definitions with primary keys, foreign keys, and data types

Submission Format:

- **File Format:** PDF document with clear headings for each task
- **Diagrams:** Include all ER diagrams (scanned hand-drawings or digital files)
- **Tables:** Use clear formatting for relational schemas
- **Page Limit:** No strict limit, but aim for clarity and conciseness

Common Mistakes to Avoid:

- Forgetting to underline primary keys in relational schemas
- Missing foreign key relationships between tables
- Stopping normalization at 2NF instead of continuing to 3NF
- ER diagrams missing cardinality constraints or participation indicators
- Weak entities without proper composite keys including owner's key
- Confusing superkeys with candidate keys

Additional Resources:

- Database textbook chapters on ER modeling and normalization
- Online ER diagram tools: Draw.io, Lucidchart, ERDPlus
- Course slides and lecture notes

Good luck!