exercise 1 students toomuch repittion

0NFStudents

Students( StudentName, StudentAddress, Age, NotYetPaid, StudentRoom, AtHome, EmailAddress1, EmailAddress2, EmailAddress3, P1BasicProgramming, P2BasicProgramming, P1DatabaseManagement, P2DatabaseManagement, P1FullStack, P2FullStack, AvgP1, AvgP2 )

1NF (First Normal Form)

**Students**

(\*\*StudentID\*\*, StudentName, StudentAddress, Age, NotYetPaid, StudentRoom, AtHome)

**StudentEmail**

(\*\*StudentID\*\*, EmailAddress)

**ExamResults**

(\*\*StudentID, CourseName, Period\*\*, Grade)

2NF

**Students**

(\*\*StudentID\*\*, Name, Address, Age, StudentRoom, AtHome)

**StudentEmail**

(\*\*StudentID, EmailAddress\*\*)

**Courses**

(\*\*CourseID\*\*, CourseName)

**ExamResults**

(\*\*StudentID, CourseID, Period\*\*, Grade)

Paid and NotYetPaid could also be in a **Payments** table if dynamic.

3NF

* \*StudentID\*\*, Name, Address, Age, StudentRoom, AtHome

**StudentEmails**

* \*StudentID, EmailAddress\*\*

**Courses**

* \*CourseID\*\*, CourseName

**Exams**

* \*StudentID, CourseID, Period\*\*, Grade

***(Optional)* Payments**

* \*StudentID\*\*, PaidAmount, RemainingAmount

#Exercise2

| **Project** | **Budget** | **Employee** | **Division** | **Director** | **Time (h)** |
| --- | --- | --- | --- | --- | --- |
| 001 | 25.000 | 003 Nathan |  |  |  |
| 005 Steven | Analyse |  |  |  |  |
| Development | Johan |  |  |  |  |
| Bram | 70 |  |  |  |  |
| 80 |  |  |  |  |  |
| 002 | 80.000 | 004 Stijn |  |  |  |
| 005 Steven | Analyse |  |  |  |  |
| Development | Johan |  |  |  |  |
| Bram | 180 |  |  |  |  |
| 30 |  |  |  |  |  |

0NF

ProjectRG(\*ProjecgNumber,\*Budget,RG(EmployeeNumber,Employeename,Division,Director,Time)) repeating group employeenmber ,EmployeeName,Division,Director Time

Split repeating group

Projects (\*projectNumber,\*Budget)

ProjectAssignement (\_ProjectNumber,EmployeeNumber\*\*,EmplyeeName,Division,Director,Time\*\*

**Exercise 2: Business Administration**

**0NF**  
**ProjectRG(\_**ProjectNumber**\_, Budget, RG(EmployeeNumber, EmployeeName, Division, Director, Time))**

* Repeating group: EmployeeNumber/EmployeeName/Division/Director/Time.

**1NF**  
Split repeating group.

* **Projects(\_**ProjectNumber**\_, Budget)**
* **ProjectAssignments(\_**ProjectNumber, EmployeeNumber**\_, EmployeeName, Division, Director, Time)**

**2NF**  
Ensure non-key attributes depend on the entire primary key. Split **ProjectAssignments**.

* **Projects(\_**ProjectNumber**\_, Budget)**
* **Employees(\_**EmployeeNumber**\_, EmployeeName, Division)**
* **ProjectAssignments(\_**ProjectNumber, EmployeeNumber**\_, Time)**
* **Divisions(\_**Division**\_, Director)**

**3NF**  
Remove transitive dependencies (none present).

* **Projects(\_**ProjectNumber**\_, Budget)**
* **Employees(\_**EmployeeNumber**\_, EmployeeName, Division)**
* **ProjectAssignments(\_**ProjectNumber, EmployeeNumber**\_, Time)**
* **Divisions(\_**Division**\_, Director)**

**Notes**:

* Corrected the original 2NF, which incorrectly included Director in **Employees**. **Divisions** separates Director to avoid redundancy.
* **ProjectAssignments** links projects and employees with time spent

**Exercise 3: Surfclub**

**0NF**  
**MemberRG(\_**MemberID**\_, MemberName, MemberHouseNo, MemberStreet, MembershipDate, RG(Surfstyle), RG(EventID, EventName, EventDate, EventPlace, EventRegion, EventPrice))**

* Repeating groups: Surfstyle, Event details.

**1NF**  
Split repeating groups.

* **Members(\_**MemberID**\_, MemberName, MemberHouseNo, MemberStreet, MembershipDate)**
* **MemberSurfStyles(\_**MemberID, Surfstyle**\_)**
* **EventParticipation(\_**MemberID, EventID**\_, EventName, EventDate, EventPlace, EventRegion, EventPrice)**

**2NF**  
Ensure non-key attributes depend on the entire primary key. Split **EventParticipation**.

* **Members(\_**MemberID**\_, MemberName, MemberHouseNo, MemberStreet, MembershipDate)**
* **MemberSurfStyles(\_**MemberID, Surfstyle**\_)**
* **Events(\_**EventID**\_, EventName, EventDate, EventPlace, EventRegion, EventPrice)**
* **EventParticipation(\_**MemberID, EventID**\_)**

**3NF**  
Remove transitive dependencies (e.g., EventRegion depends on EventPlace).

* **Members(\_**MemberID**\_, MemberName, MemberHouseNo, MemberStreet, MembershipDate)**
* **SurfStyles(\_**Surfstyle**\_)**
* **MemberSurfStyles(\_**MemberID, Surfstyle**\_)**
* **Events(\_**EventID**\_, EventName, EventDate, EventPlace, EventPrice)**
* **EventPlaces(\_**EventPlace**\_, EventRegion)**
* **EventParticipation(\_**MemberID, EventID**\_)**

**Notes**:

* Corrected original 2NF, which incorrectly used **Surfno** (not in data). **SurfStyles** is separated for flexibility.
* **EventPlaces** eliminates transitive dependency of EventRegion.

**Exercise 3: Surfclub**

**0NF**  
**MemberRG(\_**MemberID**\_, MemberName, MemberHouseNo, MemberStreet, MembershipDate, RG(Surfstyle), RG(EventID, EventName, EventDate, EventPlace, EventRegion, EventPrice))**

* Repeating groups: Surfstyle, Event details.

**1NF**  
Split repeating groups.

* **Members(\_**MemberID**\_, MemberName, MemberHouseNo, MemberStreet, MembershipDate)**
* **MemberSurfStyles(\_**MemberID, Surfstyle**\_)**
* **EventParticipation(\_**MemberID, EventID**\_, EventName, EventDate, EventPlace, EventRegion, EventPrice)**

**2NF**  
Ensure non-key attributes depend on the entire primary key. Split **EventParticipation**.

* **Members(\_**MemberID**\_, MemberName, MemberHouseNo, MemberStreet, MembershipDate)**
* **MemberSurfStyles(\_**MemberID, Surfstyle**\_)**
* **Events(\_**EventID**\_, EventName, EventDate, EventPlace, EventRegion, EventPrice)**
* **EventParticipation(\_**MemberID, EventID**\_)**

**3NF**  
Remove transitive dependencies (e.g., EventRegion depends on EventPlace).

* **Members(\_**MemberID**\_, MemberName, MemberHouseNo, MemberStreet, MembershipDate)**
* **SurfStyles(\_**Surfstyle**\_)**
* **MemberSurfStyles(\_**MemberID, Surfstyle**\_)**
* **Events(\_**EventID**\_, EventName, EventDate, EventPlace, EventPrice)**
* **EventPlaces(\_**EventPlace**\_, EventRegion)**
* **EventParticipation(\_**MemberID, EventID**\_)**

**Notes**:

* Corrected original 2NF, which incorrectly used **Surfno** (not in data). **SurfStyles** is separated for flexibility.
* **EventPlaces** eliminates transitive dependency of EventRegion.

**Exercise 4: Cookbooks**

**0NF**  
**BooksRG(\_**BookNo**\_, Title, Subtitle, Language, Website, PubNo, PubName, PubCity, PubCountry, RG(AuthorID, AuthorFirstName, AuthorLastName, AuthorCity, AuthorCountry), RG(EditionID, EditionMonth, EditionYear))**

* Repeating groups: Authors, Editions.

**1NF**  
Split repeating groups.

* **Books(\_**BookNo**\_, Title, Subtitle, Language, Website, PubNo)**
* **Publishers(\_**PubNo**\_, PubName, PubCity, PubCountry)**
* **BookAuthors(\_**BookNo, AuthorID**\_, AuthorFirstName, AuthorLastName, AuthorCity, AuthorCountry)**
* **Editions(\_**BookNo, EditionID**\_, EditionMonth, EditionYear)**

**2NF**  
Ensure non-key attributes depend on the entire primary key.

* **Books(\_**BookNo**\_, Title, Subtitle, Language, Website, PubNo)**
* **Publishers(\_**PubNo**\_, PubName, PubCity, PubCountry)**
* **Authors(\_**AuthorID**\_, AuthorFirstName, AuthorLastName, AuthorCity, AuthorCountry)**
* **BookAuthors(\_**BookNo, AuthorID**\_)**
* **Editions(\_**BookNo, EditionID**\_, EditionMonth, EditionYear)**

**3NF**  
Remove transitive dependencies (e.g., AuthorCountry depends on AuthorCity).

* **Books(\_**BookNo**\_, Title, Subtitle, Language, Website, PubNo)**
* **Publishers(\_**PubNo**\_, PubName, PubCity)**
* **Cities(\_**City**\_, Country)**
* **Authors(\_**AuthorID**\_, AuthorFirstName, AuthorLastName, AuthorCity)**
* **BookAuthors(\_**BookNo, AuthorID**\_)**
* **Editions(\_**BookNo, EditionID**\_, EditionMonth, EditionYear)**

**Notes**:

* Corrected original 3NF, which had **publicity** (likely a typo for PubCity). **Cities** separates Country.
* **BookAuthors** links books to authors, avoiding redundancy.

**Exercise 5: Home Administration**

**0NF**  
**HomeRG(\_**DepartmentNumber**\_, DepartmentName, LeaderNumber, LeaderName, MaxCapacity, RG(ChildID, ChildName, ChildBirthDate, RG(EventCode, EventType, Count)))**

* Repeating groups: Children, Events.

**1NF**  
Split repeating groups.

* **Departments(\_**DepartmentNumber**\_, DepartmentName, LeaderNumber, LeaderName, MaxCapacity)**
* **Children(\_**ChildID**\_, ChildName, ChildBirthDate, DepartmentNumber)**
* **DepartmentEvents(\_**DepartmentNumber, ChildID, EventCode**\_, EventType, Count)**

**2NF**  
Ensure non-key attributes depend on the entire primary key.

* **Departments(\_**DepartmentNumber**\_, DepartmentName, LeaderNumber, MaxCapacity)**
* **Leaders(\_**LeaderNumber**\_, LeaderName)**
* **Children(\_**ChildID**\_, ChildName, ChildBirthDate, DepartmentNumber)**
* **EventTypes(\_**EventCode**\_, EventType)**
* **DepartmentEvents(\_**DepartmentNumber, ChildID, EventCode**\_, Count)**

**3NF**  
Remove transitive dependencies (none present).

* **Departments(\_**DepartmentNumber**\_, DepartmentName, LeaderNumber, MaxCapacity)**
* **Leaders(\_**LeaderNumber**\_, LeaderName)**
* **Children(\_**ChildID**\_, ChildName, ChildBirthDate, DepartmentNumber)**
* **EventTypes(\_**EventCode**\_, EventType)**
* **DepartmentEvents(\_**DepartmentNumber, ChildID, EventCode**\_, Count)**

**Notes**:

* Corrected original 1NF, which missed splitting **LeaderName**. **Leaders** is a separate table.
* **DepartmentEvents** includes **ChildID** as events are per child per department.

**Exercise 6: Student Administration**

**0NF**  
**StudentRG(\_**StudentNo**\_, StudentName, Major, CourseNo, CourseName, InstructorNo, InstructorName, InstructorLocation, Grade)**

* Repeating groups: Course, Instructor, Grade.

**1NF**  
Split repeating groups.

* **Students(\_**StudentNo**\_, StudentName, Major)**
* **Courses(\_**CourseNo**\_, CourseName, InstructorNo)**
* **Instructors(\_**InstructorNo**\_, InstructorName, InstructorLocation)**
* **Grades(\_**StudentNo, CourseNo**\_, Grade)**

**2NF**  
Ensure non-key attributes depend on the entire primary key (already in 2NF).

* **Students(\_**StudentNo**\_, StudentName, Major)**
* **Courses(\_**CourseNo**\_, CourseName, InstructorNo)**
* **Instructors(\_**InstructorNo**\_, InstructorName, InstructorLocation)**
* **Grades(\_**StudentNo, CourseNo**\_, Grade)**

**3NF**  
Remove transitive dependencies (e.g., Major details).

* **Majors(\_**MajorCode**\_, MajorName)**
* **Students(\_**StudentNo**\_, StudentName, MajorCode)**
* **Courses(\_**CourseNo**\_, CourseName, InstructorNo)**
* **Instructors(\_**InstructorNo**\_, InstructorName, InstructorLocation)**
* **Grades(\_**StudentNo, CourseNo**\_, Grade)**

**Notes**:

* Corrected original 1NF, which combined tables unnecessarily. **Majors** separates Major details.

**Exercise 7: Patient Administration**

**0NF**  
**VisitRG(\_**VisitNo**\_, VisitDate, PatNo, PatAge, PatCity, RG(ProvNo, ProvSpecialty, Diagnosis))**

* Repeating group: Provider/Diagnosis.

**1NF**  
Split repeating group.

* **Visits(\_**VisitNo**\_, VisitDate, PatNo)**
* **Patients(\_**PatNo**\_, PatAge, PatCity)**
* **Diagnoses(\_**VisitNo, ProvNo**\_, ProvSpecialty, Diagnosis)**

**2NF**  
Ensure non-key attributes depend on the entire primary key.

* **Visits(\_**VisitNo**\_, VisitDate, PatNo)**
* **Patients(\_**PatNo**\_, PatAge, PatCity)**
* **Providers(\_**ProvNo**\_, ProvSpecialty)**
* **Diagnoses(\_**VisitNo, ProvNo**\_, Diagnosis)**

**3NF**  
Remove transitive dependencies (none present).

* **Visits(\_**VisitNo**\_, VisitDate, PatNo)**
* **Patients(\_**PatNo**\_, PatAge, PatCity)**
* **Providers(\_**ProvNo**\_, ProvSpecialty)**
* **Diagnoses(\_**VisitNo, ProvNo**\_, Diagnosis)**

**Notes**:

* Corrected original 1NF, which added **PatName, PatAddress** (not in data). **Patients** only includes provided attributes.
* **Diagnoses** links providers to visits

**Exercise 8: Dinner Club**

**0NF**  
**DinnerClubRG(\_**MemberNum**\_, MemberName, MemberAddress, DinnerNum, DinnerDate, VenueCode, VenueDescription, FoodCode, FoodDescription)**

* Repeating group: Food items.

**1NF**  
Split repeating group.

* **Members(\_**MemberNum**\_, MemberName, MemberAddress)**
* **Dinners(\_**DinnerNum**\_, DinnerDate, VenueCode)**
* **Venues(\_**VenueCode**\_, VenueDescription)**
* **Foods(\_**FoodCode**\_, FoodDescription)**
* **DinnerAttendance(\_**MemberNum, DinnerNum**\_)**
* **DinnerFoods(\_**DinnerNum, FoodCode**\_)**

**2NF**  
Ensure non-key attributes depend on the entire primary key (already in 2NF).

* **Members(\_**MemberNum**\_, MemberName, MemberAddress)**
* **Dinners(\_**DinnerNum**\_, DinnerDate, VenueCode)**
* **Venues(\_**VenueCode**\_, VenueDescription)**
* **Foods(\_**FoodCode**\_, FoodDescription)**
* **DinnerAttendance(\_**MemberNum, DinnerNum**\_)**
* **DinnerFoods(\_**DinnerNum, FoodCode**\_)**

**3NF**  
Remove transitive dependencies (none present).

* **Members(\_**MemberNum**\_, MemberName, MemberAddress)**
* **Dinners(\_**DinnerNum**\_, DinnerDate, VenueCode)**
* **Venues(\_**VenueCode**\_, VenueDescription)**
* **Foods(\_**FoodCode**\_, FoodDescription)**
* **DinnerAttendance(\_**MemberNum, DinnerNum**\_)**
* **DinnerFoods(\_**DinnerNum, FoodCode**\_)**

**Notes**:

* Corrected original 0NF, which missed **FoodCode/FoodDescription**. **DinnerFoods** links foods to dinners.
* **DinnerAttendance** ensures one dinner per member per d

**Exercise 9: Oldtimers**

**0NF**  
**ClubRG(\_**MemberID**\_, Name, Address, MembershipStart, MembershipEnd, HonorarySince, RG(Role, RoleStart, RoleEnd), RG(Brand, Model, Year, Passengers), RG(EventName, Category, Time, Location, Fee))**

* Repeating groups: Roles, Cars, Events.

**1NF**  
Split repeating groups.

* **Members(\_**MemberID**\_, Name, Address, MembershipStart, MembershipEnd, HonorarySince)**
* **MemberRoles(\_**MemberID, Role, RoleStart**\_, RoleEnd)**
* **Cars(\_**CarID**\_, MemberID, Brand, Model, Year, Passengers)**
* **Events(\_**EventID**\_, EventName, Category, Time, Location, Fee)**
* **EventParticipation(\_**EventID, MemberID, CarID**\_)**

**2NF**  
Ensure non-key attributes depend on the entire primary key (already in 2NF).

* **Members(\_**MemberID**\_, Name, Address, MembershipStart, MembershipEnd, HonorarySince)**
* **MemberRoles(\_**MemberID, Role, RoleStart**\_, RoleEnd)**
* **Cars(\_**CarID**\_, MemberID, Brand, Model, Year, Passengers)**
* **Events(\_**EventID**\_, EventName, Category, Time, Location, Fee)**
* **EventParticipation(\_**EventID, MemberID, CarID**\_)**

**3NF**  
Remove transitive dependencies (none present).

* **Members(\_**MemberID**\_, Name, Address, MembershipStart, MembershipEnd, HonorarySince)**
* **MemberRoles(\_**MemberID, Role, RoleStart**\_, RoleEnd)**
* **Cars(\_**CarID**\_, MemberID, Brand, Model, Year, Passengers)**
* **Events(\_**EventID**\_, EventName, Category, Time, Location, Fee)**
* **EventParticipation(\_**EventID, MemberID, CarID**\_)**

**Notes**:

* Corrected original 0NF, which included redundant attributes. **CarID** uniquely identifies cars.
* **EventParticipation** links events, members, and cars.